Cat. No. W14E-EN-01

SYSMAC CS-Series

CS1W-CORT21

User Defined CAN unit

OPERATION MANUAL

OMRON

CS1W-CORT21
User Defined CAN Unit
Operation Manual

Produced January 2009

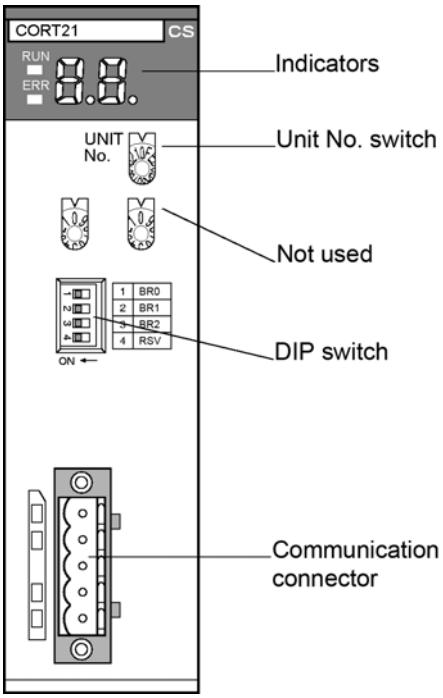
CS1W-CORT21 vs. CJ1W-CORT21

The CAN functionality of CS1W-CORT21 is identical to that of CJ1W-CORT21.

To operate the Unit, refer to attached Operation Manual W03E, and where applicable, substitute "CJ" by "CS".

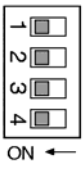
Mechanical differences between CS and CJ Units should be taken into account in the following sections of Operation Manual W03E:

2-1-1 Nomenclature and Functions



2-1-6 Switch Settings

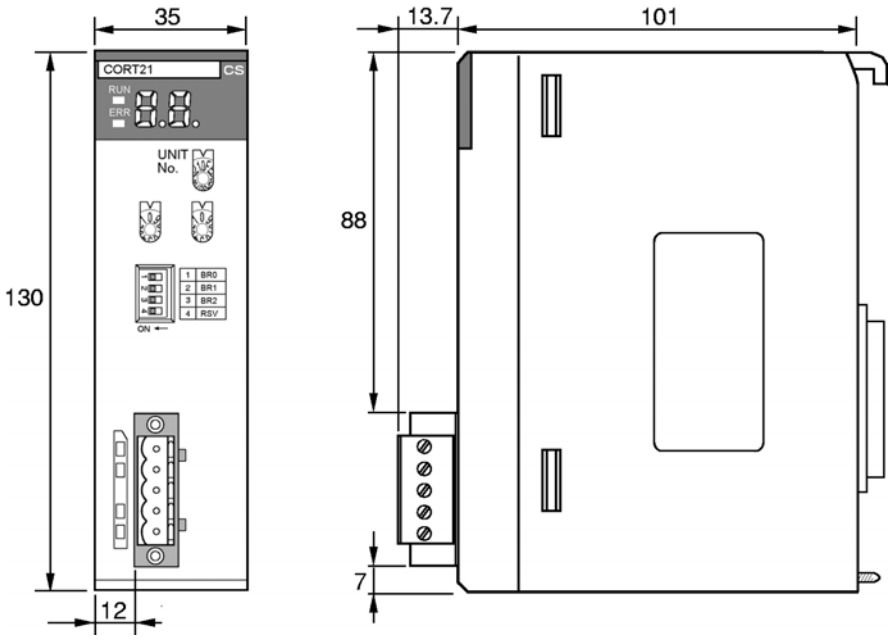
Rotary switch to set the Unit Number



Dipswitches to set the baudrate.

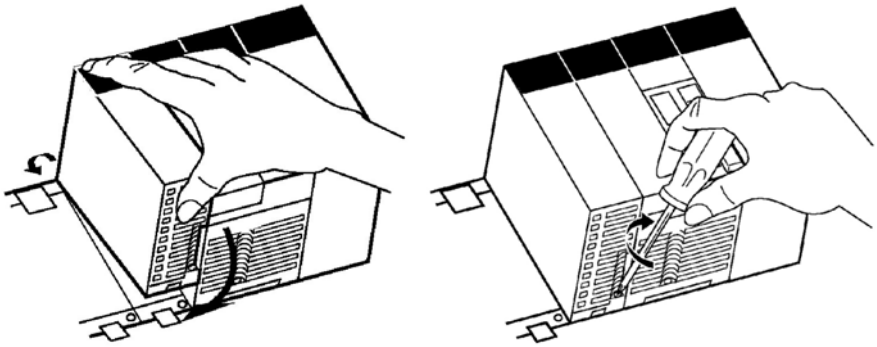
Be aware that the orientation of the dipswitches is rotated 180 degrees compared to those on the CJ1W-CORT21. The ON position is to the left.

2-2-2 External Dimensions



Dimensions are in millimeters.

2-2-3 Mounting



1. Mount the Unit to the Backplane by hooking the top of the Unit into the slot on the Backplane and rotating the I/O Unit downwards. Make sure that the connector on the back of the Unit is properly inserted into the connector in the Backplane.
2. Use a Phillips-head screwdriver to tighten* the screw on the bottom of Unit. The screwdriver must be held at a slight angle, so be sure to leave enough space below each Rack.

To remove a Unit, use a Phillips-head screwdriver to loosen the screw at the bottom of the Unit, rotate the Unit upward, and remove it.

* The screw at the bottom of the Unit must be tightened with a torque of 0.4 Nm

CJ1W-CORT21

User Defined CAN unit

OPERATION MANUAL

OMRON

CJ1W-CORT21
User Defined CAN unit

Operation Manual

Produced November 4, 2004

Notice:

OMRON products are manufactured for use by a trained operator and only for the purposes described in this manual.

The following conventions are used to classify and explain the precautions in this manual. Always heed the information provided with them.

⚠ DANGER Indicates information that, if not heeded, is likely to result in serious injury or loss of life.

⚠ WARNING Indicates information that, if not heeded, could possibly result in serious injury or loss of life.

⚠ Caution Indicates information that, if not heeded, could possibly result in minor or relatively serious injury, damage to the product or faulty operation.

OMRON Product References

All OMRON products are capitalized in this manual. The first letter of the word *Unit* is also capitalized when it refers to an OMRON product, regardless of whether it appears in the proper name of the product.

The abbreviation *Ch* appears in some displays and on some OMRON products. It often means *word* and is abbreviated as *Wd* in the documentation.

The abbreviation *PLC* means Programmable Logic Controller.

Visual Aids

The following headings appear in the left column of the manual to help you locate different types of information.

Note Indicates information of particular interest for efficient and convenient operation of the product.

Reference Indicates supplementary information on related topics that may be of interest to the user.

1, 2, 3... Indicates various lists such as procedures, checklists etc.

Trademarks and Copyrights

CAN Protocol is developed by Robert Bosch GmbH and protected by patents.

SAE is the trademark of The Society of Automotive Engineers.

CiA is the trademark of CAN in Automation(CiA), CiA is the international users' and manufacturers' organization that develops and supports CAN-based higher-layer protocols.

Other product names and company names in this manual are trademarks or registered trademarks of their respective companies.

© **OMRON, 2004**

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form, or by any means, mechanical, electronic, photocopying, recording, or otherwise, without the prior written permission of OMRON.

No patent liability is assumed with respect to the use of the information contained herein. Moreover, because OMRON is constantly striving to improve its high-quality products, the information contained in this manual is subject to change without notice. Every precaution has been taken in the preparation of this manual. Nevertheless, OMRON assumes no responsibility for errors or omissions. Neither is any liability assumed for damages resulting from the use of the information contained in this publication.

TABLE OF CONTENTS

About this Manual ix

PRECAUTIONS xi

1 Intended Audience xii
2 General Precautions xii
3 Safety Precautions xii
4 Operating Environment Precautions xiii
5 Application Precautions xiv
6 Conformance to EC Directives xvi

SECTION 1
Features and Specifications 1

1-1 Overview of CAN bus 2
1-2 User Defined CAN Unit 6
1-3 Basic Operating Procedure 9

SECTION 2
Nomenclature and Installation 11

2-1 Nomenclature and Functions 12
2-2 Installing the User Defined CAN Unit 21

SECTION 3
CAN Operation 23

3-1 Introduction 24
3-2 Physical CAN connection 25
3-3 Principles of data exchange 25
3-4 Principle of non-destructive bitwise arbitration 26
3-5 Message frames 29
3-6 Implementations of the CAN protocol 33

SECTION 4
Allocated CIO and DM Words 35

4-1 Overview of Word Allocations 36
4-2 CIO Area Words 37
4-3 Memory areas 42

SECTION 5
FINS Commands and Responses 47

5-1 Overview 48
5-2 FINS Communications 48
5-3 Using FINS Message Communications 50

TABLE OF CONTENTS

5-4 Fins Commands and Responses 54

SECTION 6
Operation 69

6-1 Performance 70
6-2 Overall performance 74

SECTION 7
Troubleshooting and Maintenance 79

7-1 Troubleshooting the User Defined CAN Unit 80
7-2 Indicators 81
7-3 Error Log Functions 84
7-4 Error responses from FINS commands 85
7-5 Using status information 91
7-6 Maintenance and Replacement 93

Abbreviations 105

Index 107

Revision History 111

About this Manual

This manual describes the installation and operation of CJ1W-CORT21, User Defined CAN Unit for SYSMAC CJ-series PLC's, and includes the sections described below.

Please read this manual carefully so that you understand the information provided before installing or using the CJ1W-CORT21 User Defined CAN Unit. Start with the precautions in the following section. They describe the operating environment and application safety measures which must be observed prior to and when using the CJ1W-CORT21 User Defined CAN unit.

The sections of this manual are as follows:

Section 1 provides an overview of CAN bus, including features, specification and system configuration of the User Defined CAN Unit.

Section 2 describes the nomenclature and installation of the User Defined CAN Unit.

Section 3 describes how CAN communication functions, and the way layer 7 protocols use CAN-communication.

Section 4 describes the words allocated to the User Defined CAN Unit in the CIO Area and DM Area. These words both enable controlling the User Defined CAN Unit and accessing Unit and bus status.

Section 5 describes message communication using FINS commands and their responses.

Section 6 describes the operation of the User Defined CAN Unit.

Section 7 describes error processing, periodic maintenance, and troubleshooting procedures needed to keep the User Defined CAN Unit operating properly. We recommend reading through the error processing procedures before operation so that operating errors can be identified and corrected more quickly.

The ***Appendices*** contain information supplementary to the information in the main body of the manual. They are referred to in the various sections as required.

The following manuals provide information on several OMRON products.

Manual	Products	Contents	Cat. No.
CJ-series Programmable controllers operation Manual	SYSMAC CJ-series CJ1G-CPU□□	Describes the installation and operation of the CJ-series PLCs.	W393-E1-□
CS/CJ-series Programmable Controllers Programming Manual	SYSMAC CS/CJ-series CS1G/H-CPU□□-E, CJ1G-CPU□□	Describes the ladder diagram programming instructions supported by CS/CJ-series PLCs.	W394-E1-□
CS/CJ-series Programmable Controllers Instructions Reference Manual	SYSMAC CS/CJ-series CS1G/H-CPU□□-E, CJ1G-CPU□□	Describes the ladder diagram programming instructions supported by CS-series and CJ-series PLCs.	W340-E1-□
CS/CJ Series Communication Commands Reference Manual	SYSMAC CS1G/H-CPU□□-E CPU Units	Describes the C-series (Host Link) and FINS communications commands used with CS/CJ-series PLCs.	W342-E1-□
CX-Programmer Operation Manual	SYSMAC WS02-CXP□□-E CX-Programmer	Provides information on how to use the CX-Programmer, programming software which supports CS1/CJ1-series PLC's.	W414-E1-□
CX-Server Run Time User Manual	CX-Server	Provides information on how to use the CX-Server communication driver software which supports CS1/CJ1-series PLC's.	W391-E2-□

Manual	Products	Contents	Cat. No.
CS/CJ-series Ethernet Units Operation Manual	SYSMAC CS/CJ-series CS1W-ETN01/ETN11 CJ1W-ETN11 Ethernet Units	Describes the installation and operation of the CS1W-ETN01 (10Base-5), CS1W-ETN11 (10Base-T), and CJ1W-ETN11 Ethernet Units.	W343-E1-□
CS/CJ-series Serial Communications Boards and Serial Communications Units Operation Manual	SYSMAC CS/CJ-series CS1W-SCB21/41, CS1W-SCU21, CJ1W-CSU41	Describes the use of Serial Communications Units and Boards to perform serial communications with external devices.	W336-E1-□
C200/CS1-series CANopen Slave Unit Operation Manual	SYSMAC C200/CS1-series C200HW-CORT21-V1	Describes the Installation and Operation of the C200HW-CORT21-V1 CANopen Slave Unit.	W904-E2-□

⚠ WARNING Failure to read and understand the information provided in this manual may result in personal injury or death, damage to the product, or product failure. Please read each section in its entirety and be sure you understand the information provided in the section and related sections before attempting any of the procedures or operations given.

PRECAUTIONS

This section provides general precautions for using the CJ1W-CORT21 User Defined CAN Unit, Programmable Logical Controllers and related devices.

The information contained in this section is important for the safe and reliable operation of the User Defined CAN Unit and Programmable Logical Controller (PLC). You must read this section and understand the information contained before attempting to set up or operate a User Defined CAN Unit as part of a PLC.

1 Intended Audience xii
2 General Precautions xii
3 Safety Precautions xii
4 Operating Environment Precautions xiii
5 Application Precautions xiv
6 Conformance to EC Directives xvi
6-1 Applicable Directives xvi
6-2 Concepts xvi
6-3 Conformance to EC Directives xvi

1 Intended Audience

This manual is intended for the following personnel, who must also have a knowledge of electrical systems (an electrical engineer or the equivalent).

- Personnel in charge of installing Factory Automation (FA) systems.
- Personnel in charge of designing FA systems.
- Personnel in charge of managing FA systems and facilities.

2 General Precautions

The user must operate the product according to the performance specifications described in the operation manuals.

Before using the product under conditions which are not described in the manual or applying the product to nuclear control systems, railroad systems, aviation systems, vehicles, combustion systems, medical equipment, amusement machines, safety equipment, and other systems, machines, and equipment that may have a serious influence on lives and property if used improperly, consult your OMRON representative.

Make sure that the ratings and performance characteristics of the product are sufficient for the systems, machines, and equipment, and be sure to provide the systems, machines, and equipment with double safety mechanisms.

This manual provides information for programming and operating OMRON User Defined CAN Unit. Be sure to read this manual before attempting to use the unit and keep this manual close at hand for reference during operation.

⚠WARNING It is extremely important that the unit is used for its specified purpose and under the specified conditions, especially in applications that can directly or indirectly affect human life. You must consult your OMRON representative before using it in a system in the above-mentioned applications.

3 Safety Precautions

⚠WARNING Never attempt to disassemble a unit or touch the inside of a unit while power is being supplied. Doing so may result in serious electrical shock or electrocution.

⚠WARNING Provide safety measures in external circuits, i.e., not in the Programmable Logical Controller (CPU Unit including associated units; referred to as PLC), in order to ensure safety in the system if an abnormality occurs due to malfunction of the PLC or another external factor affecting the PLC operation. Not doing so may result in serious accidents.

- Emergency stop circuits, interlock circuits, limit circuits, and similar safety measures must be provided in external control circuits.
- The PLC will turn OFF all outputs when its self-diagnosis function detects any error or when a severe failure alarm (FALS) instruction is executed. As a countermeasure for such errors, external safety measures must be provided to ensure safety in the system.

- The PLC outputs may remain ON or OFF due to deposition or burning of the output relays or destruction of the output transistors. As a countermeasure for such problems, external safety measures must be provided to ensure safety in the system.When the 24-VDC output (service power supply to the PLC) is overloaded or short-circuited, the voltage may drop and result in the outputs being turned OFF. As a countermeasure for such problems, external safety measures must be provided to ensure safety in the system.

⚠ **WARNING** The CPU Unit refreshes I/O even when the program is stopped (i.e., even in PROGRAM mode). Confirm safety thoroughly in advance before changing the status of any part of memory allocated to I/O Units, Special I/O Units, or CPU Bus Units. Any changes to the data allocated to any Unit may result in unexpected operation of the loads connected to the unit. Any of the following operation may result in changes to memory status.

- Transferring I/O memory data to the CPU Unit from a Programming Device.
- Changing present values in memory from a Programming Device.
- Force-setting/-resetting bits from a Programming Device.
- Transferring I/O memory files from a Memory Card or EM file memory to the CPU Unit.
- Transferring I/O memory from a host computer or from another PC on a network.

⚠ **Caution** Execute online edit only after confirming that no adverse effects will be caused by extending the cycle time. Otherwise, the input signals may not be readable.

⚠ **Caution** Confirm safety at the destination node before transferring a program to another node or changing contents of the I/O memory area. Doing either of these without confirming safety may result in injury.

4 Operating Environment Precautions

⚠ **Caution** Do not operate the unit in any of the following locations:

- Locations subject to direct sunlight.
- Locations subject to temperatures or humidities outside the range specified in the specifications.
- Locations subject to condensation as the result of severe changes in temperature.
- Locations subject to corrosive or flammable gases.
- Locations subject to dust (especially iron dust) or salt.
- Locations subject to exposure to water, oil, or chemicals.
- Locations subject to shock or vibration.

Provide proper shielding when installing in the following locations:

- Locations subject to static electricity or other sources of noise.

- Locations subject to strong electromagnetic fields.
- Locations subject to possible exposure to radiation.
- Locations near to power supply lines.

⚠ **Caution** The operating environment of the User Defined CAN Unit can have a large effect on the longevity and reliability of the system. Unsuitable operating environments can lead to malfunction, failure and other unforeseeable problems with the system. Ensure that the operating environment is within the specified conditions at installation time and remains that way during the life of the system. Follow all installation instructions and precautions provided in the operation manuals.

5 Application Precautions

Observe the following precautions when using the User Defined CAN Unit.

⚠ **WARNING** Failure to abide by the following precautions could lead to serious or possibly fatal injury. Always heed these precautions.

- Always connect to a class-3 ground (100 Ω or less) when installing the units.

⚠ **Caution** Failure to abide by the following precautions could lead to faulty operation of the unit or the system or could damage the unit or the PLC. Always heed these precautions.

- Install double safety mechanisms to ensure safety against incorrect signals that may be produced by broken signal lines or momentary power interruptions.
- When adding a new device to the network, make sure that the baud rate is the same as other nodes.
- When adding a new CAN unit to the bus, make sure that the User Defined CAN Unit is powered down, to prevent unexpected results when starting up the new node.
- Use specified communications cables.
- Do not extend connection distances beyond the ranges given in the specifications.
- Always turn OFF the power supply to the PLC, CAN units, and Communications units before attempting any of the following.
  - Mounting or dismounting the User Defined CAN Unit to CAN or the PLC bus, Power Supply Units, I/O Units, CPU Units, or any other units.
  - Assembling a unit.
  - Setting DIP switches or rotary switches.
  - Connecting or wiring the cables.
  - Connecting or disconnecting connectors.
- Be sure that the terminal blocks, connectors, memory units, expansion cables, and other items with locking devices are properly locked into place. Improper locking may result in malfunction.
- Be sure that all the mounting screws, terminal screws, unit mounting screws, and cable connector screws are tightened to the torque specified

in the relevant manuals. Incorrect tightening torque may result in malfunction.

- Leave the label attached to the unit when wiring. Removing the label may result in malfunction if foreign matter enters the unit.
- Remove the label after the completion of wiring to ensure proper heat dissipation. Leaving the label attached may result in malfunction.
- Always use the power supply voltage specified in this manual.
- Double-check all the wiring and connection of terminal blocks and connectors before mounting the units.
- Use crimp terminals for wiring. Do not connect bare stranded wires directly to terminals.
- Observe the following precautions when wiring the communications cable.
  - Separate the communications cables from the power lines or high-tension lines.
  - Do not bend the communications cables.
  - Do not pull on the communications cables.
  - Do not place heavy objects on top of the communications cables.
  - Be sure to wire communications cable inside ducts.
  - Use appropriate communications cables.
- Take appropriate measures to ensure that the specified power with the rated voltage and frequency is supplied in places where the power supply is unstable. An incorrect power supply may result in malfunction.
- Install external breakers and take other safety measures against short-circuits in external wiring. Insufficient safety measures against short-circuits may result in burning.
- Double-check all the wiring and switch settings before turning ON the power supply.
- Check the user program for proper execution before actually running it on the unit. Not checking the program may result in an unexpected operation.
- Confirm that no adverse effect will occur in the system before attempting any of the following. Not doing so may result in an unexpected operation.
  - Changing the operating mode of the PLC.
  - Force-setting/force-resetting any bit in memory.
  - Changing the present value of any word or any set value in memory.
- After replacing units, resume operation only after transferring to the new CPU Unit and/or Special I/O Units the contents of the DM Area, HR Area, and other data required for resuming operation. Not doing so may result in an unexpected operation.
- When transporting or storing the product, cover the PCB's with electrically conductive materials to prevent LSI's and IC's from being damaged by static electricity, and also keep the product within the specified storage temperature range.
- When transporting the unit, use special packing boxes and protect it from being exposed to excessive vibration or impacts during transportation.
- Do not attempt to disassemble, repair, or modify any units.

6 Conformance to EC Directives

6-1 Applicable Directives

- EMC Directive
- Low voltage directive

6-2 Concepts

OMRON units complying with EC Directives also conform to related product standards making them easier to incorporate in other units or machines. The actual products have been checked for conformity to product standards. Whether the products conform to the standards in the system used by the customer, however, must be checked by the customer.

Product related performance of OMRON units complying with EC Directives will vary depending on the configuration, wiring, and other conditions of the equipment or control panel in which OMRON devices are installed. The customer must, therefore, perform final checks to confirm that units and the overall system conforms to product standards.

A Declaration of Conformity for the unit can be requested at your OMRON representative.

6-3 Conformance to EC Directives

CAN units should be installed as follows, for the complete configuration to meet the EC directives:

1,2,3... 1. CAN units are designed for installation inside control panels. All units must be installed within control panels.

2. Use reinforced insulation or double insulation for the DC power supplies used for the communications power supply, internal circuit power supply, and the I/O power supplies.

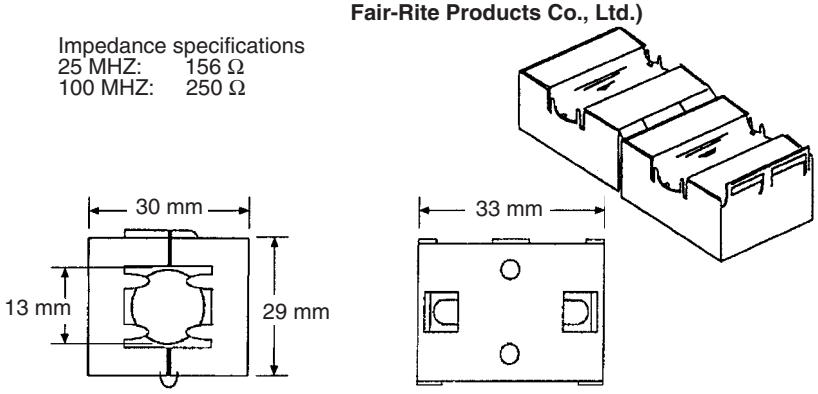
3. CAN products meet the generic emission standard. However as EMC performance can vary in the final installation, additional measures may be required to meet the standards. It should therefore be verified that the overall machine or device also meets the relevant standards.

The following examples show means of reducing noise.

1,2,3.... 1. Noise from the communications cable can be reduced by installing a ferrite core on the communications cable within 10 cm from the User Defined CAN Unit.

Ferrite Core (Data Line Filter): 0443-164151 (manufactured by Fair-Rite Products Co., Ltd.)

Impedance specifications
25 MHZ: 156 Ω
100 MHZ: 250 Ω



2. Wire the control panel with as thick and short electric lines as possible and ground to 100 Ω min.
3. Keep communications cables as short as possible and ground to 100 Ω min.

SECTION 1
Features and Specifications

This section provides an introductory overview of CAN bus, its functions and how to setup and configure a network. It also addresses the User Defined CAN Unit its configuration, features and specifications.

1-1 Overview of CAN bus 2
1-1-1 Introduction 2
1-1-2 CAN Communication Protocol 2
1-1-2-1 OSI Layer 1: Transmission Medium 3
1-1-2-2 OSI Layer 2: Datalink layer 4
1-1-2-3 OSI Layer 7: Higher layer protocol 6
1-1-3 Setting up a CAN Network 6
1-2 User Defined CAN Unit 6
1-2-1 User Defined CAN Unit Features 6
1-2-2 User Defined CAN Unit 7
1-3 Basic Operating Procedure 9
1-3-1 Overview 9
1-3-2 Preparations for Communications 10

1-1 Overview of CAN bus

1-1-1 Introduction

CAN

The Controller Area Network (CAN) is a serial bus system especially suited for networking 'intelligent' devices as well as sensors and actuators within a system or sub-system.

Application areas

CAN networks can be used as an embedded communication system for micro controllers as well as an open communication system for intelligent devices. The CAN serial bus system, originally developed for use in automobiles, is increasingly being used in industrial field bus systems, the similarities are remarkable. In both cases some of the major requirements are: low cost, the ability to function in a difficult electrical environment, a high degree of real-time capability and ease of use. Some users, for example in the field of medical engineering, opted for CAN because they have to meet particularly stringent safety requirements. Similar problems are faced by manufacturers of other equipment with very high safety or reliability requirements (e.g. robots, lifts and transportation systems).

Serial bus

CAN is a serial bus system with multi-master capabilities, that is, all CAN nodes are able to transmit data and several CAN nodes can request the bus simultaneously. The serial bus system with real-time capabilities is the subject of the ISO 11898 international standard and covers the lowest two layers of the ISO/OSI reference model. In CAN networks there is no addressing of subscribers or stations in the conventional sense, but instead, prioritized messages are transmitted.

A transmitter sends a message to all CAN nodes (broadcasting). Each node decides on the basis of the identifier received whether it should process the message or not. The identifier also determines the priority that the message enjoys in competition for bus access. The relative simplicity of the CAN protocol means that very little cost and effort need to be expended on personal training; the CAN chips interfaces make applications programming relatively simple. Introductory courses, function libraries, starter kits, host interfaces, IO modules and tools are available from a variety of vendors permitting low-cost implementation of CAN networks. Low-cost controller chips implementing the CAN data link layer protocol in silicon and permitting simple connection to micro controllers have been available since 1989. Today there are more than 50 CAN protocol controller chips from more than 15 manufacturers announced, and available.

License of CAN

The CAN protocol is developed by Robert Bosch GmbH and protected by patents.

1-1-2 CAN Communication Protocol

OSI reference model

ISO-7498

In general, the CAN communication protocol is based on the Open System Interconnection (OSI) reference model in accordance with the international standard ISO-7498 (see the following illustration). The model defines 7 layers of communication functions, two of which - layers 1 and 2 - are used in CAN.

- Layer 1, the Physical Layer of this model, defines the physical transmission characteristics.
- Layer 2, the Data Link Layer of this model, defines the bus access protocol. This protocol also includes data security and the handling of transmission protocols and telegrams.

User Interface Layer	
(7) Application Layer	Higher Layer Protocol
(6) Presentation Layer	NOT DEFINED
(5) Session Layer	
(4) Transport Layer	
(3) Network Layer	
(2) Data Link Layer	CAN communication protocol
(1) Physical Layer	CAN bus, ISO -11898

OSI Layer 1 and 2

CAN uses layers 1 and 2. Layers 3 to 7 are not defined for CAN, the application layer, OSI layer 7, defines the interface functions for specific application areas. The User Defined CAN Unit has a user defined application layer, the PLC program configures the Unit, processes received messages, collects data to transmit and triggers the Unit to send messages. This streamlined architecture ensures fast and efficient data transmission. The application functions which are available to the user, as well as the system and device behaviour of the various CAN device types, are specified in the higher layer protocol.

1-1-2-1 OSI Layer 1: Transmission Medium

ISO 11898 defines the physical layer. The CAN bus is a balanced (differential) 2-wire interface running over either a Shielded Twisted Pair (STP), Unshielded Twisted Pair (UTP), or ribbon cable. The bit encoding used is: Non Return to Zero (NRZ) encoding (with bit-stuffing) for data communication on a differential two wire bus. The use of NRZ encoding ensures compact messages with a minimum number of transitions and high resilience to external disturbance.

Serial bus

A number of different data rates are defined, with 1Mbps (Bits per second) being the top end, and 10kbps the minimum rate. Cable length depends on the data rate used. The maximum line length is 5 km, and the minimum is 25 meters at 1Mbps. Termination resistors are used at each end of the cable. The worst-case transmission time of an 8-byte frame with an 11-bit identifier is 134 bit times (that's 134 microseconds at the maximum baud rate of 1Mbits/sec).

Transmission Speed

Transmission speeds between 10 kbit/s and 1000 kbit/s can be selected as shown in the table below. One unique transmission speed must be selected for all devices on the bus when the system is commissioned

Baud rate (kbit/s)	Distance / segment (m)
10	5000
20	2500
50	1000
125	500
250	250
500	100
800	50
1000	25

Note For bus lengths greater than 1 km, a bridge or repeater device is recommended.

Cable length

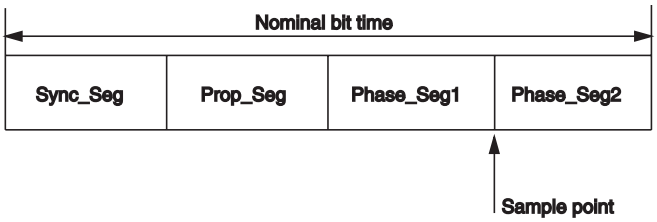
The maximum cable length values depend on the transmission speed and are based on a DeviceNet cable. The length can be increased by the use of repeaters. However, it is not recommended to use more than three repeaters in series in a CAN network.

Bit time

The User Defined CAN Unit supports bit rates in the range of 10 kBit/s up to 1000 kBit/s. Every unit in a CAN network has its own clock generator, usually a quartz oscillator. The frequency of this oscillator determines the bit rate of the unit, and therefore the reciprocal of the bit rate: the bit time. The timing parameter of the bit time (i.e. the sample point) can be configured individually for each CAN unit, creating a common bit rate even though the CAN units' oscillator periods (fosc) may be different. The nominal bit rate is the number of bits per second transmitted in the absence of resynchronisation by an ideal transmitter. The nominal bit time is the reciprocal of the nominal bitrate:

nominal bittime = 1 / nominal bit rate.

The frequencies of these oscillators are not absolutely stable, small variations are caused by changes in temperature or voltage and by deteriorating components. As long as the variations remain inside a specific oscillator tolerance range, the CAN nodes are able to compensate for the different bit rates by re synchronizing to the bit stream. According to the CAN specification, the nominal bit time is divided into four segments. The Synchronization Segment, the Propagation Time Segment, the Phase Buffer Segment 1, and the Phase Buffer Segment 2. Each segment consists of a specific, programmable number of time quanta. The length of the time quantum (t_q), which is the basic time unit of the bit time, is defined by the CAN controller's system clock f_{sys} and the Baud Rate Prescaler (BRP): t_q = BRP / f_{sys}. Typical system clocks are: f_{sys} = f_{osc} or f_{sys} = f_{osc}/2. The Synchronization Segment Sync_Seg is that part of the bit time where edges of the CAN bus level are expected to occur; the distance between an edge that occurs outside of Sync_Seg and the Sync_Seg is called the phase error of that edge. A signal-edge is expected to lie in this segment, and is used to synchronize the bus input to the system clock. The Propagation Time Segment Prop_Seg is intended to compensate for the physical delay times within the CAN network. The Phase Buffer Segments Phase_Seg1 and Phase_Seg2 surround the Sample point.



Sample point

The sample point is the point of time at which the bus level is read and interpreted as the value of that respective bit. It's location is at the end of Phase_Seg1. The sample point is set as a percentage of the total bit time: 0% - 100%. It can be set indirectly or directly for the User Defined CAN Unit. It is set indirectly with the baudrate switches on the front, or directly using a FINS command (FINS command 2909).

1-1-2-2 OSI Layer 2: Datalink layer

The CAN bus access protocol is implemented by OSI layer 2. This protocol also includes data security and the handling of the transmission protocols and messages. The datalink layer is layer 2 of the ISO/OSI reference model, and

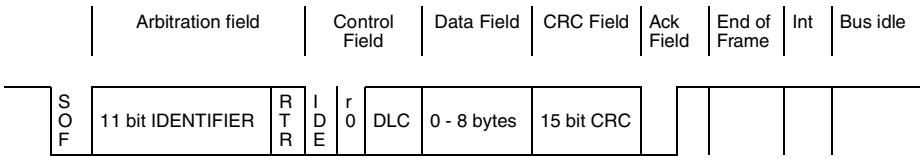
is built with two sub-layers on top of each other: the Logical Link Control (LLC) sublayer and the Medium Access Control (MAC) sublayer.

MAC

The Medium Access Control (MAC) specifies the procedures which determine when a device is permitted to transmit data. Information from transmitters to receivers is passed in data frames. The MAC sublayer represents the kernel of the CAN protocol. It presents messages received from the LLC sublayer and accepts messages to be transmitted to the LLC sublayer. The MAC sublayer is responsible for Message Framing, Arbitration, Acknowledgement, Error Detection and Signalling.

LLC

The LLC sublayer is concerned with Message Filtering, Overload Notification and Recovery Management. For a detailed description of the MAC and LLC sublayer refer to Bosch CAN Specification Version 2.0.



Data frame

The data frame is composed of an Arbitration field, Control field, Data field, CRC field, ACK field. The frame begins with a 'Start of frame' [SOF], and ends with an 'End of frame' [*EOF*] space. The data field may be from 0 to 8 bytes. The frame check sequence is derived from a Cyclic Redundancy Code (CRC); the coefficients are generated modulo-2: X15 + X14 + X10 + X8 + X7 + X4 + X3 + 1. CAN implements five error detection mechanisms; 3 at the message level and 2 at the bit level [Also incorporates error flags]. At the message level: Cyclic Redundancy Checks (CRC), Frame Checks, Acknowledgment Error Checks. At the bit level: Bit Monitoring, Bit Stuffing.

Messages

Information on the bus is sent in fixed format messages of different but limited length. Whenever the bus is free any connected unit may start to transmit a new message. In CAN systems a CAN node does not make use of any information about the system configuration (e.g. station addresses). This has several important consequences.

- System flexibility: nodes can be added to the CAN network without requiring any change in the software or hardware of any node and application layer
- Message routing: the content of a message is named by an Identifier. The Identifier does not indicate the destination of the message, but describes the meaning of the data, so that all nodes in the network are able to decide by Message Filtering whether the data is to be acted upon by then or not.
- Multicast: as a consequence of the concept of Message Filtering any number of nodes can receive and simultaneously act upon the same message.
- Data consistency: within a CAN network it is guaranteed that a message is simultaneously accepted either by all nodes or by no node. Thus data consistency of a system is achieved by the concepts of multicast and by error handling.

Message transfer

The (application) messages transmitted and received by the User Defined CAN Unit are so called Data Frames. There are two different formats for a data frame, the two formats differ in the length of the message identifier. This identifier is a field in the frame and part of the so called arbitration field, see the schematic picture of a data frame above. Data frames having identifiers with a size of 11 bits are denoted Standard Frames. The other format has

frames containing a 29 bit identifier, these frames are denoted Extended Frames. The maximum length of the data (e.g. temperature, rpm) transmitted in one message is 8 bytes. The length of data in a message can be 8 bytes or less, even if the data length is less then 8, the frame has always 8 data bytes.

Identifier length

- Standard format: the identifier's length is 11 bits and corresponds to the Base Id in the Extended format. The indentifier is interpreted as an integer value in the hexadecimal range 0x0000 - 0x07FF.
- Extended format: in contrast to the Standard format the identifier in this format consists of 29 bits. The format comprises two sections Base Id and Extended Id. The Base Id consists of 11 bits, and is the identifier for the Standard format. The Extended Id consists of 18 bits, with the Base Id this is 29 bits. The indentifier is interpreted as an integer value in the hexadecimal range 0x00000000 - 0x1FFFFFFF.

1-1-2-3 OSI Layer 7: Higher layer protocol

Examples of the higher layer protocol in OSI layer 7 are DeviceNet and J1939 (a set of standards concerning the design and use of devices that transmit and receive electronic signals and control information). For the User Defined CAN Unit the higher layer protocol is implemented in the program of the CPU in the PLC-system. Received CAN messages are stored in PLC memory areas, the CAN messages to transmit are formatted in the PLC program and (temporary) stored in PLC memory areas. The location and size of these memory area's are defined with FINS commands. Separate FINS commands can be used for direct transmission of CAN messages.

1-1-3 Setting up a CAN Network

In order to operate a CAN network, each unit in the network needs to be configured. This process of network and unit configuration involves:

- setting up the physical network topology, i.e. installing User Defined CAN Units in the PLC system, installing any other CAN unit in the network, wiring the network.
- setting up the bus parameters, which define the baud rate and the bus timing parameter sample point. Configuration for the User Defined CAN Unit is done either with the baudrate switches in the front panel, or with FINS commands.
- defining the configuration data, i.e. defining the process data, which will be exchanged between the User Defined CAN Unit and other nodes on the CAN network.
- defining the parameterization data for the User Defined CAN Unit, which defines the filtering of message identifiers, the configuration of message buffers in the PLC system.
- setting up the parameterization of the User Defined CAN Unit, with FINS commands issued from the program in the PLC system.

1-2 User Defined CAN Unit

1-2-1 User Defined CAN Unit Features

User Defined CAN Unit

The User Defined CAN Unit (CJ1W-CORT21) is a CPU Bus Unit, which can be installed on a CJ1 PLC System. The User Defined CAN Unit will interface between the Central Processor Unit (CPU) of the PLC system and the CAN bus. The User Defined CAN Unit is connected to the CPU of the PLC system through the backplane bus of the CJ1 system, this backplane bus is also noted as the PC21 bus. The physical connection is made via a CJ1 main rack

or a CJ1 expansion rack. The User Defined CAN Unit is connected to the CAN bus with a 5-pin open style connector.

CPU Bus Unit

A total of up to 16 CPU Bus Units can be mounted on the CPU Rack or an Expansion Rack. The total of 16 must include all User Defined CAN Units and all other CPU Bus Units

Unit Control and Status

Up to 25 words of control and status words are exchanged between the User Defined CAN Unit and a dedicated CIO memory area, of which the location is related to the unit number. Control bits, allow the PLC program to switch the CAN communication of the Unit between enabled and disabled, communication over the CAN bus is only active in the enabled mode. During the disabled mode no communication will take place from the Unit to the CAN network, or vice versa.

The remaining CIO words provide status and diagnostics information on the Unit itself, the CAN network and the state of messages.

DM Area

A total of 100 DM words is allocated for every User Defined CAN Unit placed in the PLC System. These allocated DM words are all reserved for future use, and currently not used. That is no data is exchanged between an allocated DM area and the User Defined CAN Unit. As these words are reserved it is not recommended to use these areas in the PLC program.

FINS Messages

The User Defined CAN Unit supports Factory Intelligent Network Services (FINS) message exchange with the CPU of the PLC System, four types of services are supported: configuration, transmission, error log and identification services.

Configuration

Required for normal operation of CAN network is an external 24V power supply, connected to the Unit using the front connector. Before the User Defined CAN Unit is able to operate in a CAN network, the baudrate and sample point must be configured either using the selector switches on the front of the Unit, or FINS command 2909.

Troubleshooting Functions

The User Defined CAN Unit is provided with a variety of troubleshooting functions for prompt recovery in case of errors:

- Extensive self-diagnostic function at startup.
- Communication exchange flags, indicating if message buffers are exchanged with the bus-device(s).
- Status and error flags, indicating the status of the Unit and the CAN network
- Error log for recording error history data

1-2-2 User Defined CAN Unit

Model

Applicable PLC	Unit classification	Types of communications	Model number
CJ Series	CPU Bus Unit	• Message communications	CJ1W-CORT21

General Specifications

General specifications of the User Defined CAN Unit conform to the general specifications for the SYSMAC CJ-series CPU Units.

Functional and Performance Specifications

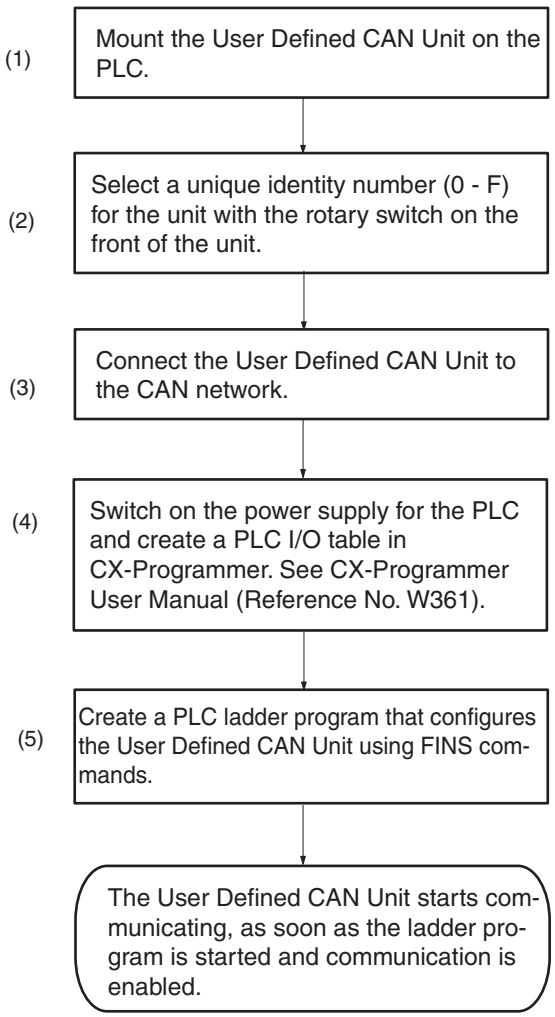
Item	Specification
Unit model	CJ1W-CORT21

Item		Specification
Applicable PLC		CJ Series
Unit classification		CPU Bus Unit
Applicable unit numbers		0 to F
Mounting position		CPU Rack or Expansion Rack
No. of Units that can be mounted		16 Units max.
No. of Units that can be connected per CAN network		Unlimited
Words allocated in the CPU Unit	CIO Area words allocated for the CPU Bus Unit	25 words/Unit (allocation for one Unit)
		CPU Unit to User Defined CAN Unit: 10 words for the software switches and status words, the remaining 15 words are reserved for future use.
	DM Area words allocated for the CPU Bus Unit	100 words/Unit (allocation for one Unit).
		Reserved, currently not used.
	Other I/O memory	Input, Output buffers, flag areas (configured with FINS commands).
Supported connections (communications)		• Message buffers and FINS message communications. All conform to CAN communications standards.
Message communications	FINS message configuration	Configuration of Input and Output buffers, with a maximum of 640 for each type. Configuration of send and receive flags area, there can be only one of each type. Configuration of baudrate and sample point.
	FINS message communications	FINS commands to send/receive directly a CAN message
Other functions	Error history in the User Defined CAN Unit	Supported. (The history can be accessed using a FINS command.)
Setting section		Rotary switches: Unit No. (hexadecimal x 1). Front panel DIP switch: Baud rate (digital x 3).
Display section		Two LED indicators: One green indicator to indicate the Unit is powered, and one red indictor to indicate an error status. Two-digit 7-segment display: Displays CAN communication status or error code. 2 dot LED indicators, in 2-digit 7-segment display: communication status, left dot indicates FINS configuration send from CPU to unit, right dot indicates communication is configured and enabled.
Front connector		One communications connector (communications data: CAN H and CAN L, network power supply: CAN_V+, ground CAN_GND and shielded) Use the XW4B-05C1-H1-D connector provided to connect the communications cable. **Note** Use the XW4B-05C4-T-D connector sold separately for multi-drop connections.
Network power supply voltage		11 to 25 VDC (supplied from the communications connector)
Influence on CPU Unit cycle time		0.3 ms+0.0064 ms x the number of buffers allocated
Current consumption		Network power supply: 18 mA at 24 VDC, (supplied from the communications connector) Internal circuit power supply: 290 mA max. at 5 VDC (supplied from the Power Supply Unit)
External dimensions		31 x 90 x 65 mm (W x H x D)
Weight		118 g (including the connector provided)
Standard accessories		One XW4B-05C1-H1-D connector to connect to a node from a T-branch Tap.

1-3 Basic Operating Procedure

1-3-1 Overview

The following diagram provides an overview of the installation procedures. For experienced installation engineers, this may provide sufficient information. For others, cross-references are made to various sections of this manual where more explicit information is given. When reading this manual online, the flow chart entries provide links to the sections containing detailed information.


(1)
Mount the User Defined CAN Unit on the PLC.
(2)
Select a unique identity number (0 - F) for the unit with the rotary switch on the front of the unit.
(3)
Connect the User Defined CAN Unit to the CAN network.
(4)
Switch on the power supply for the PLC and create a PLC I/O table in CX-Programmer. See CX-Programmer User Manual (Reference No. W361).
(5)
Create a PLC ladder program that configures the User Defined CAN Unit using FINS commands.
The User Defined CAN Unit starts communicating, as soon as the ladder program is started and communication is enabled.

1-3-2 Preparations for Communications

1,2,3...

1. Wire the network, to connect the User Defined CAN Unit to the CAN network.
1. Mount the User Defined CAN Unit on the PLC system.
   - Treat the Unit as a CPU Bus Unit.
   - It can be mounted to a CPU Rack or Expansion Rack.
   - Number of Units: 16 (Max).
2. Set the Unit No. (UNIT No.) for the User Defined CAN Unit.
3. Connect and turn ON the power supply to the PLC.
4. Connect and turn ON the 24 V power supply to the User Defined CAN Unit.
5. Create and download the PLC program to configure the User Defined CAN Unit with FINS messages.

SECTION 2
Nomenclature and Installation

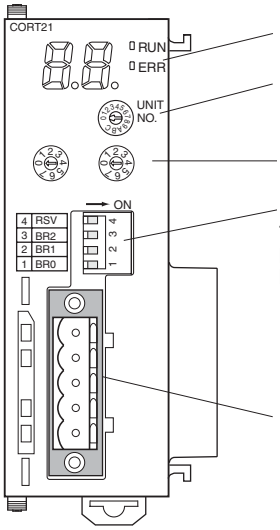
This section describes the nomenclature and installation of the User Defined CAN Unit.

2-1 Nomenclature and Functions 12
2-1-1 Nomenclature and Functions 12
2-1-2 Unit states 12
2-1-3 CAN 15
2-1-4 Status Indicators 17
2-1-5 Seven-Segment Display 17
2-1-6 Switch Settings 19
2-2 Installing the User Defined CAN Unit 21
2-2-1 System Configuration Precautions 21
2-2-2 External Dimensions 21
2-2-3 Mounting 21
2-2-4 Handling Precautions 22

2-1 Nomenclature and Functions

2-1-1 Nomenclature and Functions

CJ1W-CORT21



Indicators

Unit No. switch

This switch sets the unit number of the User Defined CAN Unit as a one-digit hexadecimal value.

These switches are reserved for future use

DIP switch

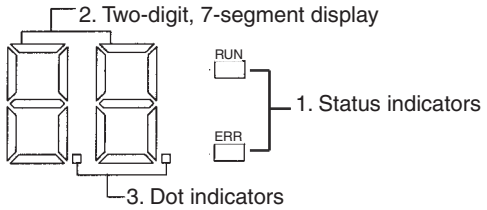
The pins have the following functions:
Pins 1, 2 and 3: Baud rate settings
Pin 4: reserved for future use

Communications connector

Connect the Network communications cable to this connector. The communications power for this Unit is also supplied through this connector.
A parallel connector with screws (XW4B-O5C1-H1-D) is provided for node connection.

Indicators

The User Defined CAN Units are equipped with the following indicators that indicate the operating status of the node itself and the overall network.



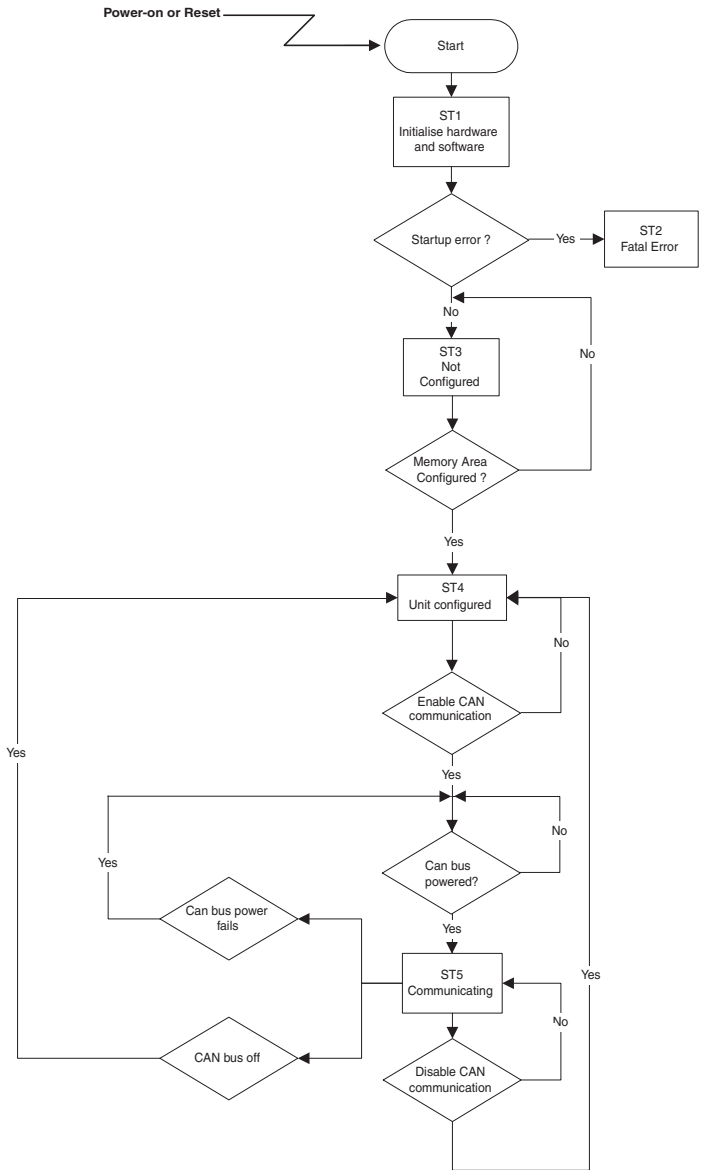
1,2,3...

1. Two status indicators, two-color: green or red LED.
2. A two-digit, 7-segment display
3. Two dot indicators (located on the lower right of each 7-segment display)

2-1-2 Unit states

The User Defined CAN Unit has several states, these states are visualized in the figure below. Events will force the Unit to switch to a different state, these events can be user initialized or caused by system behaviour (or error conditions detected in the system)

The different states are defined and described in the first table below, in the second table the events are defined which will force the Unit into a different state. In normal operating mode the Unit is initialized, and no start-up error occurred, the states are refreshed during every Unit Cyclic refresh. Changed state information is reflected as changed bits in the CIO area, the location of the CIO area is set with the unit number.



States

State	Name	Cyclic refresh FINS	Description
ST1	Initializing	No	The Unit executes start-up tests and initializes the PLC backplane bus communications
ST2	Start-up error	No	A start-up error was detected during start up tests or during the initialization of the PLC backplane communication bus.
ST3	Not configured	Yes	The Unit started without start up errors, but is not (yet) configured using FINS commands.
ST4	Configured	Yes	The Unit has received the memory locations and a number of output and input message buffers.
ST5	Communicating	Yes	CAN messages can be sent and received, if CAN network is powered.

Events

Event	Name	Description
EV1	Configure memory areas	The memory areas for message buffers, send triggers and receive flags have been successfully configured through FINS commands.
EV2	Configure one CAN message	One of the output message buffers or input message buffers has been successfully configured through FINS commands.
EV3	Enable communications	CAN messaging has been successfully enabled with CIO bits.
EV4	Disable communications	CAN messaging has been disabled with CIO bits.
EV5	Bus-off	CAN messaging has been transitioned to the bus off state, e.g. a bus-error occurred.

Unit Cyclic refresh

The User Defined CAN Unit performs a cyclic refresh for five memory areas in the CPU of the PLC system, into which it is mounted. In one of these memory ares, control and status flags are present, the state information is included in these flags.

The Unit determines independently when to cyclic refresh it's data. Several PLC Cyclic refreshes may occur between two Unit Cyclic refreshes! This means that rising/ falling edges and/or data changes must be handled with care!

During an Unit Cyclic refresh, data is transferred between the PLC and the Unit. This data is NOT processed immediately after the Unit Cyclic refresh. Several Unit Cyclic refreshes may occur between the processing of the data! This means that rising/ falling edges and/or data changes must be handled with care!

Item	Location	Description
Control and Status Flags	CIO n n = 1500 + 25 * unit number	Flags to control the Unit behavior and flags to show the status of the Unit.
Send Triggers	configurable	Triggers to send CAN messages on demand. Bits in the send trigger area will send a message if the bit-value has a rising edge and the bit is associated with an output buffer. FINS commands are used to define a send trigger area, to associate an identifier with an output buffer and a send trigger.
Receive Flags	configurable	Flags indicating which messages have been received. A received message is identified by its identifier, and in case a bit in the receive flags area is associated with the received identifier, this bit is set. The message content is placed in the corresponding input buffer. FINS commands are used to define a receive flag area, to associate an identifier with an input buffer and a receive flag.
CAN Output Message Buffers	configurable	The message content to send, every output buffer can only have one message.
CAN Input Message Buffers	configurable	The content of the received messages every input buffer can have only one message at a time. New incoming messages will overwrite the previous content of the input buffer.

FINS services

The User Defined CAN Unit supports the types of FINS services listed in the following table.

FINS Service Type	Description
Configuration Services	Services to: • Configure the configurable memory locations, number of input messages and number of output messages. • Set the identifiers and way of sending the output messages. • Set the identifiers of the input messages to handle. • Set the bit rate of the CAN physical layer.
Transmission Services	Services to send a specific output message.
Error Log Services	Services to read and clear the error log.
Identification Services	Services to identify the Unit and its firmware version.

2-1-3 CAN

CAN functionality

The User Defined CAN Unit supports two functions for messages: sending and receiving. Every message in CAN communication has an identifier, the options for these message identifiers are:

- Identifier is individually configurable, the size is 29-bit or 11-bit for each output message buffer.
- Configurable identifier for every input message buffer. All messages use either 11-bit or 29-bit length identifiers.

Sending CAN messages

FINS commands are used to define the CAN messages for the User Defined CAN Unit. These FINS commands define which messages can be send by the Unit, the mode of sending, and the times for which messages are send. Each output message buffer can have one of the three modes SM1, SM2 or SM3. And an output message must have the correct length, being less or equal to 8.

Mode	Name	Description
SM1	Triggered	Sending an output message is triggered via a bit in the PLC memory. This bit is located in the Send Triggers area.
SM2	On-change	Sending an output message is triggered as soon as the Unit detects that the message contents have changed or the message length has changed. (An incorrect message length (> 8) is not considered a change)
SM3	Cyclic	Sending an output message is triggered as soon as a specified time has elapsed.

Triggered

In send mode SM1 a rising edge in the Send Triggers area determines that an output message will be sent. In state ST5, the Unit will evaluate the trigger. In all other states a rising edge in the Send Triggers area is ignored, messages will not be send.

On-change

In send mode SM2 a change in the message content determines that the output message will be sent. In state ST5, the Unit will evaluate the message content. In all other states message contents changes are ignored, messages will not be send.

Cyclic

In send mode SM3, time determines that an output message is sent. In state ST5, the Unit checks whether the configured cycle time for the output message has elapsed. As soon as it elapses, the message is sent. In all other states are the time intervals not monitored, no messages are sent.

It is allowed and possible to combine any of the three send modes mentioned above, even all three modes can be active at the same time.

Receiving CAN messages

The User Defined CAN Unit will process any CAN message it receives, the basic processing is to retrieve the message identifier from the message and compare it with the identifiers defined for the Unit. In total there are four different scenarios whenever a message is processed.

Condition	Unit processing
The integrity check for the message fails, the check is done by the CAN controller directly, or the firmware.	The User Defined CAN Unit sends an error frame on the CAN bus. The message is rejected by the CAN controller and/or the firmware, no notification is send to the application layer.
The integrity check for the message is successful, the User Defined CAN Unit is not able to process the message due to overload conditions.	The User Defined CAN Unit sends an acknowledge on the CAN bus, ignores the message, and in the next cyclic refresh of the CPU bit 7 in CIO word n+3 is set.
The integrity check for the message is successful and the User Defined CAN Unit is able to process the message, but no input message buffer is configured for the received identifier.	The User Defined CAN Unit sends an acknowledge on the CAN bus, and ignores the message.
The integrity check for the message is successful, the User Defined CAN Unit is able to process the message and an input message buffer is configured for the received indentifier.	The User Defined CAN Unit sends an acknowledge on the CAN bus, and processes the message with the following actions in next Unit Cyclic refresh: 1. Set bit 3 in CIO word n+3. 2. The message content is copied to the first input message buffer that is configured for the received identifier. 3. Set the receive flag of the first input message buffer that is configured for the received identifier.

Back-to-back messaging, sending or receiving messages with a high rate, is not explicitly supported nor implemented.

Bus off

In case the User Defined CAN Unit detects an abnormal rate of errors on the bus, the Unit will go in ‘bus off’. This means that all CAN communication is disabled, the Unit will try to go online after the user enables the communication (normally this is initiated in the PLC program). In the ‘bus off’ state the Unit will:

1. Set CIO n + 3, bit 10
2. Fire event EV5, i.e. disable communications.
3. Switch on the ERR LED.

2-1-4 Status Indicators

The status indicators on the front panel of the User Defined CAN Unit are labelled RUN and ERR. The RUN and ERR LED indicates the status of the node itself and/or the status of the network.

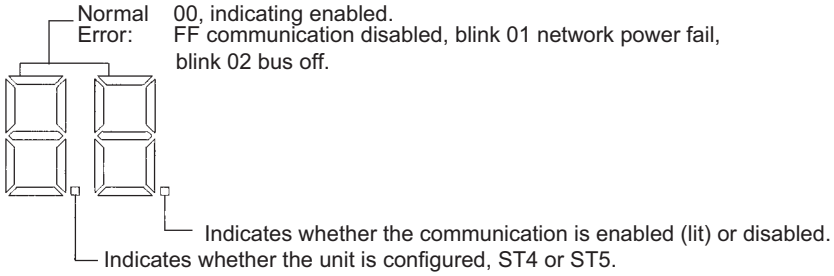
The RUN and ERR indicators can be green or red and they can be turned OFF or ON. The following table lists the different states and colours of the indicators with their meaning.

Indicator	Color	Status	Condition
RUN	Green	ON	Normal operating status, the Unit state is ST3, ST4 of ST5.
	Red	ON	1. A non-recoverable, fatal error has occurred. 2. Unit state is ST2, start-up error occurred.
	---	OFF	Power is not being supplied or the Unit is in state ST1 or ST2.
ERR	Red	ON	A fatal communication error has occurred. Network communications are not possible. One or more of the following errors active: 1. Hardware error at startup (Unit in ST2). 2. No CAN configuration. 3. CAN network power fail (bit 9 of CIO word n+3 is on). 4. Unit is Bus off (bit 10 of CIO word n+3 is on). 5. Fatal error in Unit. 6. CPU Watchdog Time-out. 7. PC21 bus error. 8. Cyclic refresh time-out. 9. I/O table error.
	---	OFF	The Unit has no active error, diagnostic functions do not detect any error condition.

2-1-5 Seven-Segment Display

In addition to the RUN and ERR indicators, User Defined CAN Units have a 2-digit, 7-segment display that normally indicates the enabled or disabled status of the communication. When an error occurs, the display will display a (flashing) error code.

There are dot indicators in the lower-right corner of each digit. The left dot of these two indicators shows whether or not the Unit is configured with FINS commands, that is ST4 or ST5. The right dot indicator shows whether the communication is enabled or disabled, the dot is lit when the communication is enabled, that is ST5.


Normal 00, indicating enabled.
Error: FF communication disabled, blink 01 network power fail,
blink 02 bus off.
Indicates whether the communication is enabled (lit) or disabled.
Indicates whether the unit is configured, ST4 or ST5.

Seven-segment Digits

The following table shows the functions of the 7-segment digits.

Status	Display
• Unit not mounted in PLC-rack. • ST1 or ST2.	Not lit.
• Unit not configured, ST3. • CAN communications are disabled, ST4.	Lit: FF
CAN communications are enabled, ST5.	Lit: 00
CAN network power failure.	Blink: 01
CAN bus off.	Blink: 02

Dot Indicators

The following table shows the functions of the dot indicators.

Indicator	Content	Display
Left dot	Unit operational	ON: Unit is configured, and communication is enabled or disabled, state is ST4 or ST5. OFF: Unit state is ST1, ST2 or ST3, Unit has a start-up error or is not (yet) configured. BLINK: Unit state is ST2.
Right dot	Communication enabled/disabled	ON: Communication is enabled, ST5 (left is dot is lit). OFF: Communication is disabled, if left dot is lit, otherwise Unit is in state ST1, ST2 or ST3.

Normal start-up

During a normal start-up the indicators will show the following information:

Action	Indicator	State
Power OFF State = ST0	RUN	-
	ERR	-
	7-Segment	--
	Left dot	-
	Right dot	-
Power ON State = ST3	RUN	Green
	ERR	-
	7-Segment	FF
	Left dot	-
	Right dot	-
Areas configured, FINS 2902 State = ST4	RUN	Green
	ERR	-
	7-Segment	FF
	Left dot	Red
	Right dot	-
Buffer(s) configured, FINS 2903 - 2906 State = ST4	RUN	Green
	ERR	-
	7-Segment	FF
	Left dot	Red
	Right dot	-

Action	Indicator	State
Communication enabled State = ST5	RUN	Green
	ERR	-
	7-Segment	00
	Left dot	Red
	Right dot	Red
Communication disabled State = ST4	RUN	OFF
	ERR	-
	7-Segment	FF
	Left dot	Red
	Right dot	-

2-1-6 Switch Settings

Unit No. Switch

The unit number is used to identify individual CPU Bus Units when more than one CPU Bus Unit is mounted to the same PLC. The unit number must be unique for each CPU Bus Unit. Selecting a non-unique number for a CPU Bus Unit will prevent the PLC System from starting correctly. Use this switch to set the unit number of the User Defined CAN Unit as a CPU Bus Unit. The unit number setting determines the CIO and DM area words allocated to the Unit as software switches and the status area.

Always turn OFF the power to the PLC CPU before changing the unit number setting. The Unit only reads the Unit number setting during the initialisation following a power-up, but not following a software reset.

Setting method: One-digit hexadecimal

Setting range: 0 to F

Note

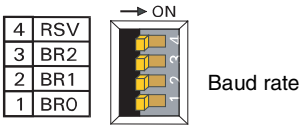
1. The unit number is set to 0 at the factory.
2. Any unit number from 0 to F can be set as long as it hasn't been set on another CPU Bus Unit connected to the same PLC.
3. Use a small flat-blade screwdriver to turn the rotary switches; be careful not to damage the switch.
4. Always turn OFF the PLC before changing the unit number setting.
5. If the unit number is the same as one set on another CPU Bus Unit connected to the same PLC, a duplicate number error will occur in the PLC and it won't be possible to start up the CAN network.
6. If the unit number is being set for the first time or changed, then an I/O table may be created for the PLC.
7. The two 8-states rotary switch are reserved for future use.

Rotary switches

The two rotary switches on the front, below the unit no. switch, are not used, and reserved for future use.

DIP Switch

The DIP switches on the front of the User Defined CAN Unit are used to set the baud rate, and implicitly the sample point. The baudrate is set with switches 1, 2, and 3, switch number 4 is spare and currently not used.



The settings of the DIP switch pins for the baudrate are shown in the following table. All pins are set to OFF at the factory.

.

DIP-switch			Baud rate (kbit/s)	Sample point (%)
1	2	3		
OFF	OFF	OFF	10	80
ON	OFF	OFF	20	80
OFF	ON	OFF	50	80
ON	ON	OFF	125	80
OFF	OFF	ON	250	80
ON	OFF	ON	500	80
OFF	ON	ON	reserved	reserved
ON	ON	ON	1000	70[1]

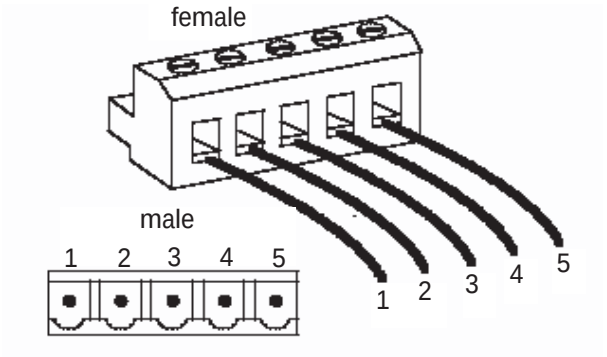
1. Sample point is reduced due to the clock frequency.

Setting the switch

1. Always turn OFF the PLC before changing the DIP switch settings.
2. Set the same baud rate on all of the nodes in the network. Any unit with a baud rate different from the other unit(s)'s baud rate won't be able to participate in communications and may cause a communications error between units that have been set properly.

2-1-7 CAN Bus connector

The CAN bus connector is a 5-pin open style connector. This CAN connector is located on the front side of the Unit, and is a male connector with five pins.



Color stickers that match communication cable colors are attached to the communications connectors. Match the colors when connecting communication cables to the connectors, the colors can be found in the next table.

Pin no.	Signal	Color	Description
1	CAN_GND	Black	Ground
2	CAN_L	Blue	Communications line, dominant low
3	CAN_SHLD	----	Shield, optional
4	CAN_H	White	Communications line, dominant high
5	CAN_V+	Red	External positive power supply 24V, for powering the transceiver and optocouplers

Note Before connecting communications cables, turn OFF the PLC power supply, and the communications power supply.

2-2 Installing the User Defined CAN Unit

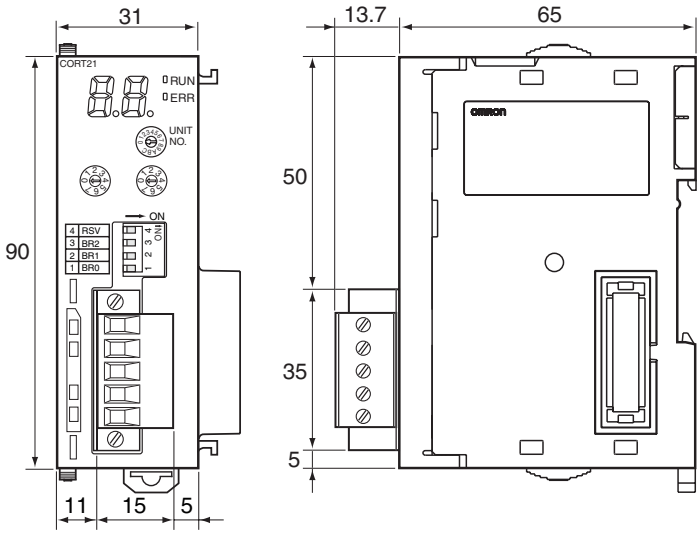
2-2-1 System Configuration Precautions

- I/O words are allocated to CPU Bus Units according to the unit number setting on the switch located on the front panel of the Unit, not according to unit slot numbers. Refer to *4-1 Overview of Word Allocations.*
- In the CJ-series, up to 16 units can be mounted to the CPU main rack or expansion rack (but no more than 10 units on either).

2-2-2 External Dimensions

CJ1W-CORT21

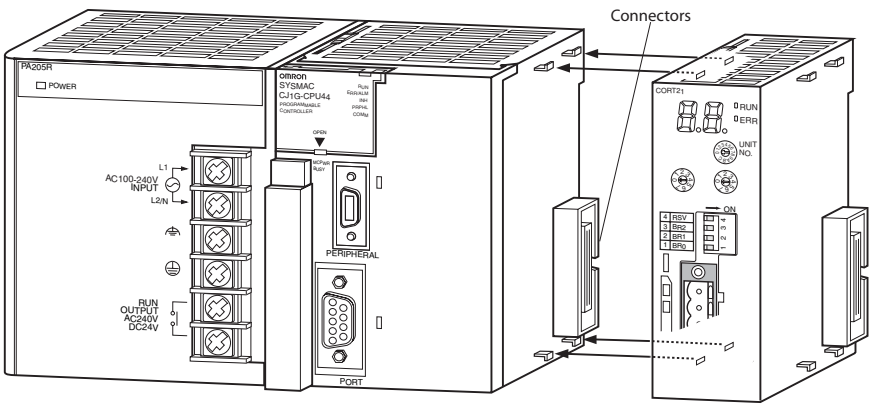
This diagram in this paragraph shows the dimensions of the User Defined CAN Unit. Refer to the CJ series CPU Unit Operation Manual (W393) for the dimensions of the Unit when it is mounted to the PLC bus. All dimensions are in mm.



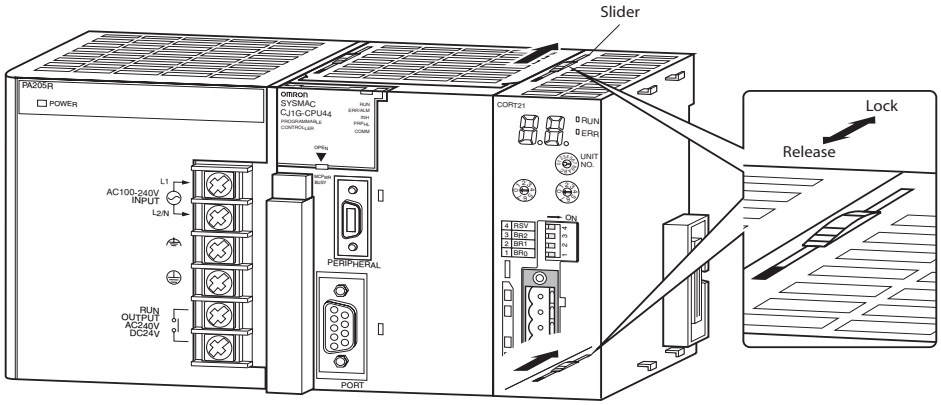
2-2-3 Mounting

CJ-series PLC

1. Carefully align the connectors to mount the User Defined CAN Unit.



2. Move the yellow sliders on the top and bottom of the Unit until they click into position, to lock.

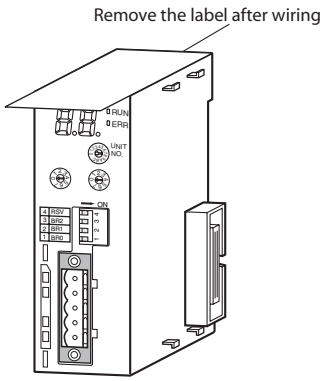


Note If the sliders are not securely locked, the User Defined CAN Unit functions may not operate sufficiently.

To dismount the Unit, move the sliders to the “Release” direction.

2-2-4 Handling Precautions

- Always turn OFF the power supply to the PLC before mounting or dismounting a Unit or connecting or disconnecting cables.
- Provide separate conduits or ducts for the I/O lines to prevent noise from high-tension lines or power lines.
- Leave the label attached to the Unit when wiring. Removing the label may result in malfunction if foreign matter enters the Unit.
- Remove the label after the completion of wiring to ensure proper heat dissipation. Leaving the label attached may result in malfunction.



SECTION 3
CAN Operation

This section describes how to send and receive messages with the User Defined CAN Unit. It will discuss what needs to be configured for the unit, and how CAN messages are processed, and will explain the principles of CAN communication.

3-1 Introduction 24
3-2 Physical CAN connection 25
3-3 Principles of data exchange 25
3-4 Principle of non-destructive bitwise arbitration 26
3-4-1 Non-destructive bitwise arbitration 26
3-4-2 Efficiency of bus allocation 27
3-5 Message frames 29
3-5-1 Message frame formats 29
3-5-2 Detecting and signalling errors 30
3-5-3 Data reliability of the CAN protocol 31
3-5-4 Extended format CAN message 32
3-6 Implementations of the CAN protocol 33
Overview 33
CAN controller with intermediate buffer 33
CAN controller with object storage 33
CAN slave controllers for I/O functions 33

3-1 Introduction

Controller Area Network

The Controller Area Network (CAN) is a serial communications protocol which efficiently supports distributed real-time control with a very high level of security.Its domain of application ranges from high speed networks to low cost multiplex wiring.
In automotive electronics, engine control units, sensors, anti-skid-systems, etc. are connected using CAN with bitrates up to 1 Mbit/s. At the same time it is cost effective to build into vehicle body electronics, e.g. lamp clusters, electric windows etc. to replace the wiring harness otherwise required. CAN has the following properties:

1. prioritization of messages
2. guarantee of latency times
3. configuration flexibility
4. multicast reception with time synchronization
5. system wide data consistency
6. multi master
7. error detection and signalling
8. automatic retransmission of corrupted messages as soon as the bus is idle again
9. distinction between temporary errors and permanent failures of nodes
10. autonomous switching off of defect nodes

To achieve design transparency and implementation flexibility CAN has been subdivided into different layers according to the ISO/OSI Reference Model:

Layer 7	**Application Layer**
Layer 3 - 6	
Layer 2	**Data Link Layer**
	Logical Link Control: LLC Acceptance filtering overload notification recovery management Medium Access Control: MAC Data encapsulation and decapsulation frame coding, stuffing medium access management error detection error signalling acknowledgement serialization, deseriliazation
Layer 1	**Physical Layer**
	bit encoding/decoding bit timing synchronization

Physical Layer

The Physical Layer defines how signals are actually transmitted, and is this layer are bit timing, bit encoding and synchronization handled.

MAC sublayer

The MAC sublayer is the kernel of the CAN protocol, messages received from the LLC sub layer are processed, and messages from the physical layer are

send to the MAC sublayer. In the MAC sublayer the message framing, arbitration, acknowledgement, error detection and signalling are handled.

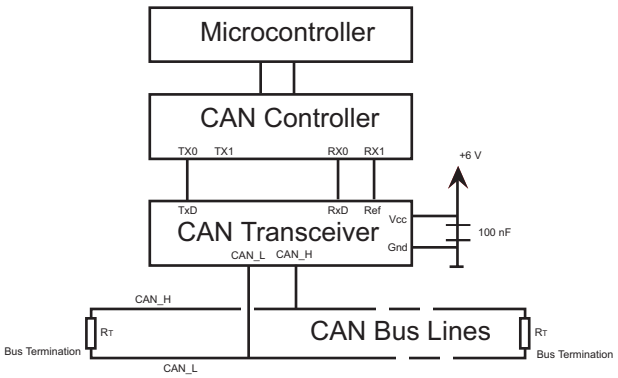
LLC sublayer

The LLC sublayer does the following processing on messages received/transmitted from/to the MAC sublayer: message filtering. It also deals with overload notification and recovery management.

3-2 Physical CAN connection

ISO 11898

The data rates (up to 1 Mbit/s) necessitate a sufficiently steep pulse slope, which can be implemented only by using power elements. A number of physical connections are basically possible. However, the users and manufacturers group CAN in Automation recommends the use of driver circuits in accordance with ISO 11898. Integrated driver chips in accordance with ISO 11898 are available from several companies. The international users and manufacturers group (CiA) also specifies several mechanical connections (cable and connectors).



3-3 Principles of data exchange

Unit addressing

When data are transmitted by CAN, no units are addressed, but instead, the content of the message (e.g. rpm or engine temperature) is designated by an identifier that is unique throughout the network. The identifier defines not only the content but also the priority of the message. This is important for bus allocation when several units are competing for bus access.

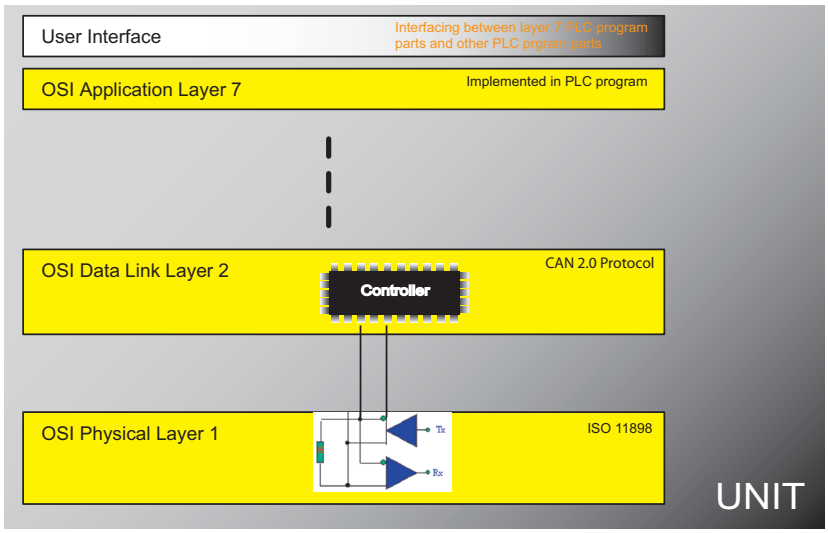
Sending messages

If the CPU of a given unit wishes to send a message to one or more units, it passes the data to be transmitted and their identifiers to the assigned CAN chip. This is all the CPU has to do: To initiate data exchange. The message is constructed and transmitted by the CAN chip. As soon as the CAN chip receives the bus allocation all other units on the CAN network become receivers of this message. Each unit in the CAN network, having received the message correctly, performs an acceptance test to determine whether the data received are relevant for that unit. If the data are of significance for the unit concerned they are processed, otherwise they are ignored.

Addressing scheme

A high degree of system and configuration flexibility is achieved as a result of the content-oriented addressing scheme. It is very easy to add units to the existing CAN network without making any hardware or software modifications to the existing units, provided that the new units are purely receivers. Because the data transmission protocol does not require physical destination

addresses for the individual components, it supports the concept of modular electronics and also permits multiple reception (broadcast, multicast) and the synchronization of distributed processes: measurements needed as information by several controllers can be transmitted via the network, in such a way that it is unnecessary for each controller to have its own sensor.



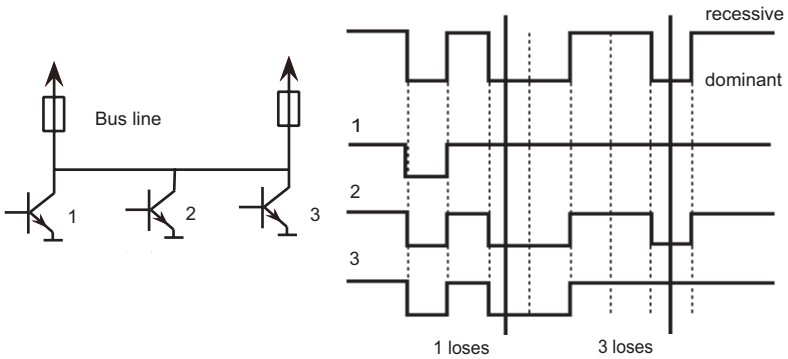
3-4 Principle of non-destructive bitwise arbitration

3-4-1 Non-destructive bitwise arbitration

For the data to be processed in real time they must be transmitted rapidly. This not only requires a physical data transfer path with up to 1 Mbit/s but also calls for rapid bus allocation when several units wish to send messages simultaneously.

Priority

In real-time processing the urgency of messages to be exchanged over the network can differ greatly: a rapidly changing dimension (e.g. engine load) has to be transmitted more frequently and therefore with less delays than other dimensions (e.g. engine temperature) which change relatively slowly.



The priority at which a message is transmitted compared with another less urgent message is specified by the identifier of the message concerned. The priorities are laid down during system design in the form of corresponding binary values and cannot be changed dynamically. The identifier with the lowest binary number has the highest priority.

Bitwise arbitration

When ever the bus is free, any unit may start to transmit a message. Bus access conflicts are resolved by bitwise arbitration on the identifiers involved by each unit observing the bus level bit for bit. In accordance with the "wired and" mechanism, by which the dominant state (logical 0) overwrites the recessive state (logical 1), the competition for bus allocation is lost by all those units with recessive transmission and dominant observation. During arbitration every transmitter compares the level of the bit transmitted with the level that is motored on the bus. If these levels are equal the unit may continue to send. All "losers" (levels are not equal) automatically become receivers of the message with the highest priority and do not re-attempt transmission until the bus is available again.

3-4-2 Efficiency of bus allocation

Bus allocation

The efficiency of the bus allocation system is determined mainly by the possible applications for a serial bus system. In order to judge as simply as possibly which bus systems are suitable for which applications the literature includes a method of classifying bus allocation procedures. Generally we distinguish between the following classes:

- Bus allocation on a fixed time schedule.
- Bus allocation on the basis of need.

Bus allocation on a fixed time schedule

Bus allocation done on a fixed time schedule, is made sequentially to each participant for a maximum duration regardless of whether this participant needs the bus at this moment or not. Examples of this type of bus allocation are: token slot or token passing.

Bus allocation on the basis of need

Allocation on the basis of need is defined as: the bus is allocated to one participant on the basis of outstanding transmission requests, i.e. the allocation system only considers participants wishing to transmit. Examples of this type of bus allocation are: CSMA, CSMA/CD, flying master, round robin or bitwise arbitration.

CAN bus allocation

For CAN, bus allocation is negotiated purely among the messages waiting to be transmitted. This means that the procedure specified by CAN is classified as allocation on the basis of need.

Bus access

Another means of assessing the efficiency of bus arbitration systems is the bus access method:

- Non-destructive bus access
- Destructive bus allocation

Non-destructive bus access

With methods of this type the bus is allocated to one and only one unit either immediately or within a specified time following a single bus access (by one or more units). This ensures that each bus access by one or more units leads to an unambiguous bus allocation. Examples of non-destructive bus access are: token slot, token passing, round robin, bitwise arbitration.

Destructive bus allocation

Simultaneous bus access by more than one unit causes all transmission attempts to be aborted and therefore there is no successful bus allocation. More than one bus access may be necessary in order to allocate the bus at all, the number of attempts before bus allocation is successful being a purely statistical quantity. Examples of destructive bus access are: CSMA/CD, Ethernet.

Unambiguous bus allocation

In order to process all transmission requests of a CAN network while complying with latency constraints at as low a data transfer rate as possible, the CAN protocol must implement a bus allocation method that guarantees that there is always unambiguous bus allocation even when there are simultaneous bus accesses from different units. The method of bitwise arbitration using the identifier of the messages to be transmitted uniquely resolves any collision between a number of units wanting to transmit, and it does this at the latest within 13 (standard format) or 33 (extended format) bit periods for any bus access period. Unlike the message-wise arbitration employed by the CSMA/CD method this non-destructive method of conflict resolution ensures that no bus capacity is used without transmitting useful information.

Even in situations where the bus is overloaded the linkage of the bus access priority to the content of the message proves to be a beneficial system attribute compared with existing CSMA/CD or token protocols: In spite of the insufficient bus transport capacity, all outstanding transmission requests are processed in order of their importance to the overall system (as determined by the message priority). The available transmission capacity is utilized efficiently for the transmission of useful data since "gaps" in bus allocation are kept very small. The collapse of the whole transmission system due to overload, as can occur with the CSMA/CD protocol, is not possible with CAN. Thus, CAN permits implementation of fast, traffic-dependent bus access which is non-destructive because of bitwise arbitration based on the message priority employed.

Bus access control

Non-destructive bus access can be further classified into

- centralized bus access control
- decentralized bus access control

depending on whether the control mechanisms are present in the system only once (centralized) or more than once (decentralized). A communication system with a designated unit (inter alia for centralized bus access control) must provide a strategy to take effect in the event of a failure of the master unit. This concept has the disadvantage that the strategy for failure management is difficult and costly to implement and also that the takeover of the central unit by a redundant unit can be very time-consuming. For these reasons and to circumvent the problem of the reliability of the master unit (and thus of the whole communication system), the CAN protocol implements decentralized bus control. All major communication mechanisms, including bus access control, are implemented several times in the system, because this is the only way to fulfill the high requirements for the availability of the communication system.

In summary it can be said that CAN implements a traffic-dependent bus allocation system that permits, by means of a non-destructive bus access with decentralized bus access control, a high useful data rate at the lowest possible bus data rate in terms of the bus busy rate for all units. The efficiency of the bus arbitration procedure is increased by the fact that the bus is utilized only by those units with pending transmission requests.

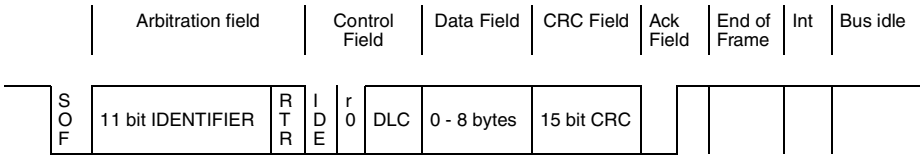
These requests are handled in the order of the importance of the messages for the system as a whole. This proves especially advantageous in overload situations. Since bus access is prioritized on the basis of the messages, it is possible to guarantee low individual latency times in real-time systems.

3-5 Message frames

3-5-1 Message frame formats

Identifier

The CAN protocol supports two message frame formats, the only essential difference being in the length of the identifier (ID). In the standard format the length of the ID is 11 bits and in the extended format the length is 29 bits. The message frame for transmitting messages on the bus comprises seven main fields.



Arbitration field

A message in the standard format begins with the start bit, Start Of Frame, this is followed by the arbitration field, which contains the identifier and the RTR (Remote Transmission Request) bit, which indicates whether it is a data frame or a request frame without any data bytes (remote frame).

Control Field

The control field contains the IDE (Identifier Extension) bit, which indicates either standard format or extended format, a bit reserved for future extensions and - in the last 4 bits - a count of the data bytes in the data field. The number of bytes in the data field is indicated by the Data Length Code (DLC), the data length code is 4 bits wide

Number of data bytes	Data Length Code			
	DLC3	DLC2	DLC1	DLC0
0	d	d	d	d
1	d	d	d	r
2	d	d	d	r
3	d	d	r	d
4	d	r	r	r
5	d	r	d	d
6	d	r	r	d
7	d	r	r	r
8	r	d	d	d

d = dominant
r = recessive

Data field

The data field ranges from 0 to 8 bytes in length and is followed by the CRC field, which is used as a frame security check for detecting bit errors.

CRC Field

The CRC field contains the CRC sequence, followed by a delimiter. this CTC-delimiter is a single recessive bit. The frame check sequence is derived from a cyclic redundancy code best suited for frames with bit counts less then 127 bits (BCH code). In order to carry out the CRC calculation the polynomial to be divided is defined as the polynomial, the coefficients of which are given by the destuffed bit stream consisting of Start Of Frame (SOF), arbitration field, control field, data field and for the 15 lowest coefficients, by 0. This polynomial is divided (the coefficients are calculated modulo-2) by the generator-polynomial:

$$X^{15} + X^{14} + X^{10} + X^{8} + X^{7} + X^{4} + X^{3} + 1$$

The remainder of this polynomial division is the CRC sequence transmitted over the bus.

Ack Field

The ACK field comprises the ACK slot (1 bit) and the ACK delimiter (1 recessive bit). The bit in the ACK slot is sent as a recessive bit and is overwritten as

a dominant bit by those receivers which have at this time received the data correctly (positive acknowledgement). Correct messages are acknowledged by the receivers regardless of the result of the acceptance test.

Intermission

The end of the message is indicated by End Of Frame. Intermission is the minimum number of bit periods separating consecutive messages. If there is no following bus access by any unit, the bus remains idle (bus idle).

3-5-2 Detecting and signalling errors

Error detection

Unlike other bus systems, the CAN protocol does not use acknowledgement messages but instead signals any errors that occur. For error detection the CAN protocol implements three mechanisms at the message level:

1.Cyclic Redundancy Check (CRC)
The CRC safeguards the information in the frame by adding redundant check bits at the transmission end. At the receiver end these bits are re-computed and tested against the received bits. If they do not agree there has been a CRC error.

2.Frame check
This mechanism verifies the structure of the transmitted frame by checking the bit fields against the fixed format and the frame size. Errors detected by frame checks are designated "format errors".

3.ACK errors
As mentioned above, frames received are acknowledged by all recipients through positive acknowledgement. If no acknowledgement is received by the transmitter of the message (ACK error) this may mean that there is a transmission error which has been detected only by the recipients, that the ACK field has been corrupted or that there are no receivers.

Bit level errors

The CAN protocol also implements two mechanisms for error detection at the bit level:

1.Monitoring
The ability of the transmitter to detect errors is based on the monitoring of bus signals: each node which transmits also observes the bus level and thus detects differences between the bit sent and the bit received. This permits reliable detection of all global errors and errors local to the transmitter.

2.Bit stuffing
The coding of the individual bits is tested at bit level. The bit representation used by CAN is NRZ (non-return-to-zero) coding, which guarantees maximum efficiency in bit coding. The synchronization edges are generated by means of bit stuffing, i.e. after five consecutive equal bits the sender inserts into the bit stream a stuff bit with the complementary value, which is removed by the receivers. The code check is limited to checking adherence to the stuffing rule.

Error flag

If one or more errors are discovered by at least one unit (any unit) using the above mechanisms, the current transmission is aborted by sending an Error Flag. This prevents other units accepting the message and thus ensures the consistency of data throughout the network.

Re-transmission

After transmission of an erroneous message has been aborted, the sender automatically re-attempts transmission (automatic repeat request). There may again be competition for bus allocation. As a rule, retransmission will begin

within 23 bit periods after error detection; in special cases the system recovery time is 31 bit periods.

However effective and efficient the method described may be, in the event of a defective unit it might lead to all messages (including correct ones) being aborted, thus blocking the bus system if no measures for self-monitoring were taken. The CAN protocol therefore provides a mechanism for distinguishing sporadic errors from permanent errors and localizing unit failures (fault confinement).

This is done by statistical assessment of unit error situations with the aim of recognizing a unit's own defects and possibly entering an operating mode where the rest of the CAN network is not negatively affected. This may go as far as the unit switching itself off to prevent messages erroneously recognized as incorrect from being aborted.

Bus off

With respect to fault confinement a CAN unit is in one of three states, the number of the state increases with the severity of the error:

1. error active
2. error passive
3. bus off

An 'error active' unit can normally take part in bus communication and sends an Active Error Flag when an error has been detected.

An 'error passive' unit must not send an Active Error Flag. It takes part in bus communication but when an error has been detected only a Passive Error Flag is sent. Also after a transmission, an 'error passive' unit will wait before initiating a further transmission.

A 'bus off' unit is not allowed to have any influence on the bus. (E.g. output drivers switched off.)

Error counters

For fault confinement two counters are implemented in every CAN unit:

- Transmit error counter
- Receive error counter

For respectively a transmit error and a receive error the corresponding counter increases, normally the receive counter is increased by one for every error, the transmit error counter is normally increased by eight. See the CAN bus specification for a detailed description of increasing the error counters, and the exceptions in increasing the counters.

A CAN unit is 'error passive' when the transmit error counter equals or exceeds 128, or when the receive error counter equals or exceeds 128. A CAN unit is 'bus off' when the transmit error counter is greater than or equal to 256.

Error reset

In 'bus off' mode, the CAN unit has its communication disabled, the ladder logic in the CPU of the PLC-system can enable the communication using a FINS command. Enabling the communication will reset both the receive and transmit counter.

3-5-3 Data reliability of the CAN protocol

Reliability

The introduction of safety-related systems in automobiles brought with it high requirements for the reliability of data transmission. The objective is frequently formulated as not permitting any dangerous situations for the driver to occur as a result of data exchange throughout the whole life of a vehicle.

This goal is achieved if the reliability of the data is sufficiently high or the residual error probability is sufficiently low. In the context of bus systems data, reliability is understood as the capability to identify data corrupted by transmission faults. The residual error probability is a statistical measure of the impairment of data reliability: It specifies the probability that data will be corrupted and that this corruption will remain undetected. The residual error probability should be so small that on average no corrupted data will go undetected throughout the whole life of a system.

Residual error

Calculation of the residual error probability requires that:

- the errors which occur are classified, and
- that the whole transmission path is described by a model.

This calculation of the residual error probability of CAN results in a maximum bit error probability, which is approximately 0.02 - in the order of 10^{-13} ($0.02 * 10^{-13)}$. The residual error probability of CAN is determined as a function of:

- the bit error probability for message lengths of 80 to 90 bits,
- for system configurations of, for instance, five or ten nodes
- and with an error rate of 1/1000 (an error in one message in every thousand).

Based on the residual error probability it is possible to calculate the maximum number of undetectable errors for a given CAN network.

Undetectable errors

For example, if a CAN network operates at a data rate of 1 Mbit/s, at an average bus capacity utilization of 50 percent, for a total operating life of 4000 hours and with an average message length of 80 bits, then the total number of messages transmitted is: 9×10^{10}.
The statistical number of undetected transmission errors during the operating life is thus in the order of less than 10^{-2}.
Or to put it another way, with an operating time of eight hours per day on 365 days per year and an error rate of 0.7 s, one undetected error occurs every thousand years (statistical average).

3-5-4 Extended format CAN message

29-bit identifier

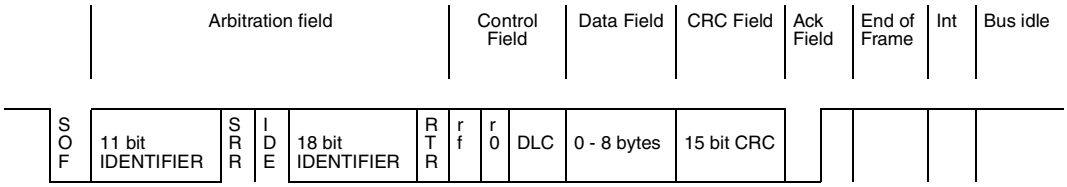
The Society of Automotive Engineers (SAE) "Truck and Bus" subcommittee standardized signals and messages as well as data transmission protocols for various data rates. It became apparent that standardization of this kind is easier to implement when a longer identification field is available.

To support these efforts, the CAN protocol was extended by the introduction of a 29-bit identifier. This identifier is made up of the existing 11-bit identifier (base ID) and an 18-bit extension (ID extension). Thus the CAN protocol allows the use of two message formats: StandardCAN (Version 2.0A) and ExtendedCAN (Version 2.0B). As the two formats have to coexist on one bus it is laid down which message has higher priority on the bus in the case of bus access collisions with differing formats and the same base identifier: The message in standard always has priority over the message in extended format.

Extended format

CAN controllers which support the messages in extended format can also send and receive messages in standard format. When CAN controllers which only cover the standard format (Version 2.0A) are used on one network, then only messages in standard format can be transmitted on the entire network. Messages in extended format would be misunderstood. However there are

CAN controllers which only support standard format but recognize messages in extended format and ignore them (Version 2.0B passive).



The distinction between standard format and extended format is made using the IDE bit (Identifier Extension Bit) which is transmitted as dominant in the case of a frame in standard format. For frames in extended format it is recessive.

RTR bit

The RTR bit is transmitted dominant or recessive depending on whether data are being transmitted or whether a specific message is being requested from a unit. In place of the RTR bit in standard format the SRR (substitute remote request) bit is transmitted for frames with extended ID. The SRR bit is always transmitted as recessive, to ensure that in the case of arbitration the standard frame always has priority bus allocation over an extended frame when both messages have the same base identifier.

IDE bit

Unlike the standard format, in the extended format the IDE bit is followed by the 18-bit ID extension, the RTR bit and a reserved bit (r1).

All the following fields are identical with standard format. Conformity between the two formats is ensured by the fact that the CAN controllers which support the extended format can also communicate in standard format.

3-6 Implementations of the CAN protocol

Overview

Communication is identical for all implementations of the CAN protocol. There are differences, however, with regard to the extent to which the implementation takes over message transmission from the micro controllers which follow it in the circuit.

CAN controller with intermediate buffer

CAN controllers with intermediate buffer, formerly called basicCAN chips, have implemented as hardware the logic necessary to create and verify the bitstream according to the protocol. However, the administration of messages to be sent and received, acceptance filtering in particular, is carried out only to a limited extent by the CAN controller.

CAN controller with object storage

CAN objects consist mainly of three components: identifier, data length code and the actual useful data. CAN controllers with object storage, formerly called FullCAN, function like CAN controllers with intermediate buffers. Additional they administer a certain number of objects. Where there are several simultaneous requests to transmit objects over the bus, the CAN controllers determine, for example, which object is to be transmitted first. They also carry out acceptance filtering for incoming objects. The micro controller following the CAN controller has to administer only a few bits (e.g. transmission request).

CAN slave controllers for I/O functions

As well as CAN controllers which support all functions of the CAN protocol there are also CAN implementations possible which do not require a following micro controller. These CAN implementations are called SLIO (Serial Link I/O) acting as CAN slave units and having to be administered by a CAN master unit.

SECTION 4
Allocated CIO and DM Words

This section describes the words allocated to the User Defined CAN Unit in the CIO Area and DM Area. However DM words are not used by the User Defined CAN unit.

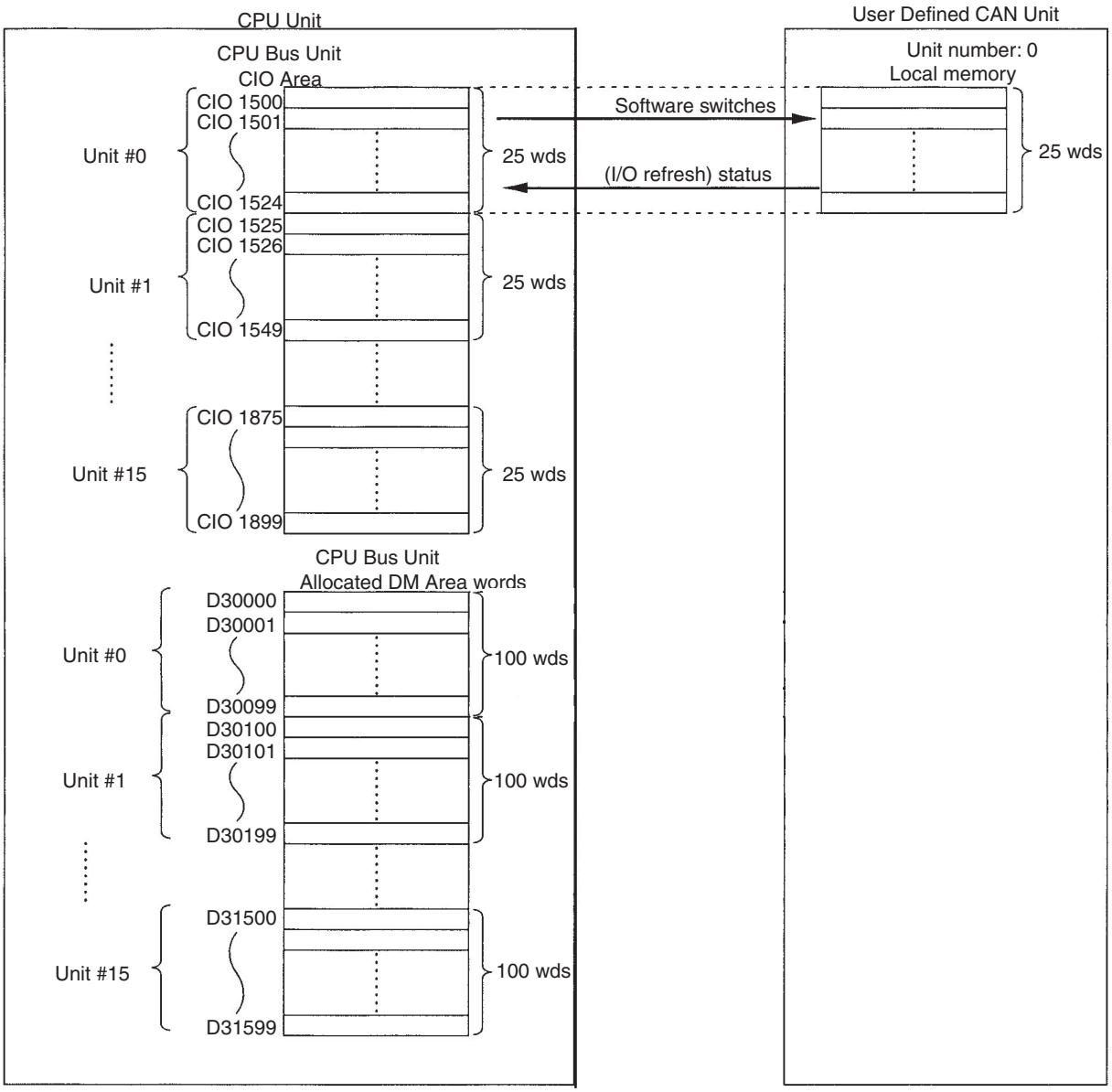
4-1 Overview of Word Allocations 36
4-1-1 CIO Area Words 37
4-1-2 Allocated DM Area Words 37
4-2 CIO Area Words 37
4-2-1 Communication enable (Word n) 39
4-2-2 Reserved output word (Words n+1, n+2) 39
4-2-3 Status communication (Word n+3) 40
4-2-4 Always zero input words (Words n+4, n+5, n+6, n+7) 41
4-2-5 Number of delayed send messages (Word n+8) 41
4-2-6 Number of received messages waiting to be processed (Word n+9) 41
4-2-7 Reserved words for the system (Words n+10 - n+24) 41
4-3 Memory areas 42
4-3-1 Send Trigger Area 42
4-3-2 Receive Flags Area 43
4-3-3 CAN Output Message Buffers 43
4-3-4 CAN Input Message Buffers 44

4-1 Overview of Word Allocations

The words shown in the following diagram are allocated according to the unit number setting. For each CPU Bus Unit, there are 25 words allocated in the CIO Area and 100 words allocated in the DM Area.

First word allocated in the CIO Area: n = CIO 1,500 + (25 x unit number)

First word allocated in the DM Area: m = DM30,000 + (100 x unit number)



4-1-1 CIO Area Words

Control flags and User Defined CAN Unit status are allocated in the CIO Area according to the unit number, as shown below. Control Flags are bits used as commands from the PLC CPU to the User Defined CAN Unit to enable executing unit functions.

Unit number	Allocated words
0	CIO 1500 to CIO 1524
1	CIO 1525 to CIO 1549
2	CIO 1550 to CIO 1574
3	CIO 1575 to CIO 1599
4	CIO 1600 to CIO 1624
5	CIO 1625 to CIO 1649
6	CIO 1650 to CIO 1674
7	CIO 1675 to CIO 1699
8	CIO 1700 to CIO 1724
9	CIO 1725 to CIO 1749
10	CIO 1750 to CIO 1774
11	CIO 1775 to CIO 1799
12	CIO 1800 to CIO 1824
13	CIO 1825 to CIO 1849
14	CIO 1850 to CIO 1874
15	CIO 1875 to CIO 1899

The unit number can be set by rotating the rotary switch at the front side of the User Defined CAN Unit. It defines the special CPU unit number on the PLC bus, the unit number is in the range 0x0 - 0xF.

4-1-2 Allocated DM Area Words

The DM area words that are allocated for the User Defined CAN Unit are not used, i.e. no data is exchanged between an allocated DM area and the unit. However, the allocated area is reserved for use in a future extension of the unit. Therefore, using this area for user data is not recommended

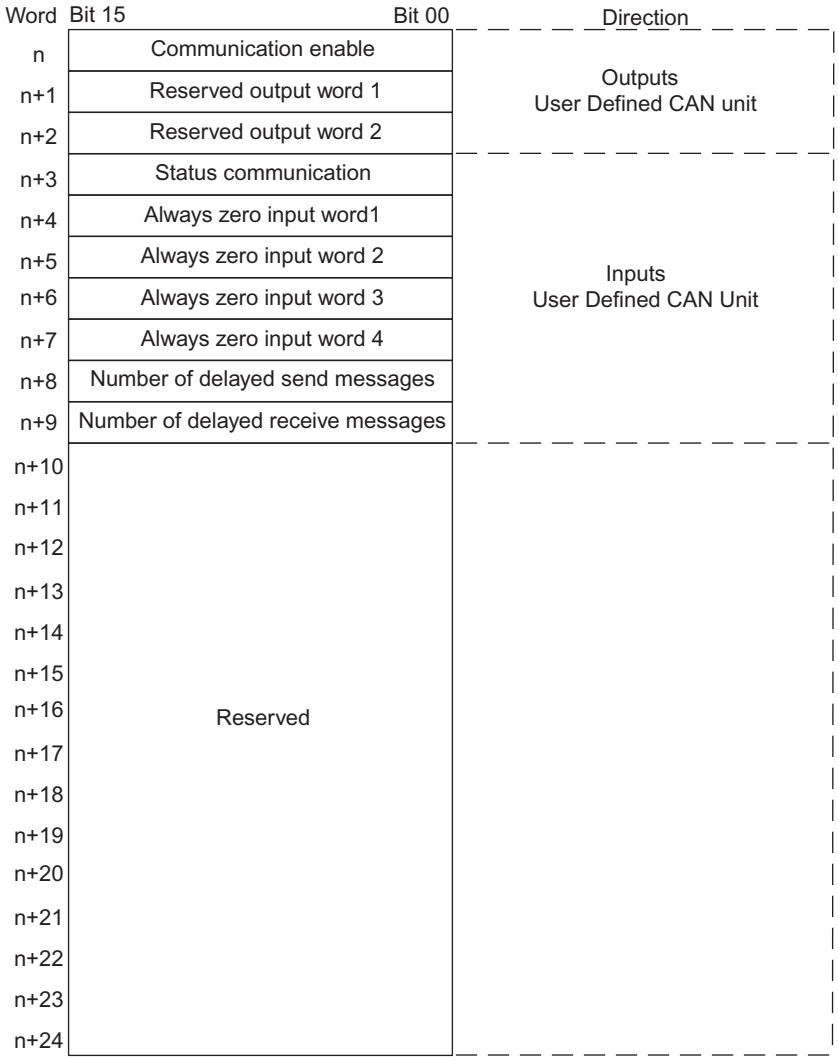
4-2 CIO Area Words

For each CPU Bus unit, the CS/CJ-series PLC allocates up to 25 words in the CIO area. Data for one unit is stored in this block of 25 words, the start address of the this block has an offset from the absolute start address CIO 1500. This offset is defined with the unit number, the unit number is set with the rotary switches on the front of the User Defined CAN Unit.

The first word of the block in the CIO area can be calculated as follows from the unit number setting:

First word n = CIO 1,500 + (25 x unit number)

The CIO area words contain control and status flags, the next chapters explain in detail these control and status flags.


Word
Bit 15
Bit 00
Direction
n
Communication enable
n+1
Reserved output word 1
n+2
Reserved output word 2
Outputs
User Defined CAN unit
n+3
Status communication
n+4
Always zero input word1
n+5
Always zero input word 2
n+6
Always zero input word 3
n+7
Always zero input word 4
n+8
Number of delayed send messages
n+9
Number of delayed receive messages
Inputs
User Defined CAN Unit
n+10
n+11
n+12
n+13
n+14
n+15
n+16
n+17
n+18
n+19
n+20
n+21
n+22
n+23
n+24
Reserved

4-2-1 Communication enable (Word n)

The User Defined CAN Unit's control bits are located in the first CIO word allocated via the unit number setting.

Word n				[n = CIO 1,500 + (25 x unit number)]
Bit	**Name**	**Status**	**Controlled by**	**Unit operation**
00-03	Reserved for the system	---	---	---
04	Enable CAN communications	ON→OFF	User	In state ST5 (communicating) 1. generate event EV4 (disable communications) 2. reset CIO n + 3, bit 6 3. reset all send triggers in other states, no action
		OFF→ON		In state ST1 (initializing), ST3 (not configured), generate error ER9 (Network parameter file lost) in state ST5 (communicating), no action in state ST4 (configured) 1. generate event EV3 (enable communications) 2. reset CIO n + 3, bit 7 3. reset CIO n + 3, bit 10 4. reset all receive flags.
05-15	Reserved for the system	---	---	---

4-2-2 Reserved output word (Words n+1, n+2)

Word n + 1, n + 2				[n = CIO 1,500 + (25 x unit number)]
Bit	**Name**	**Status**	**Controlled by**	**Unit operation**
00-15	Reserved for the system	---	---	---

4-2-3 Status communication (Word n+3)

This word give the status of the CAN bus and the User Defined CAN Unit. Also it is possible to enable the communication.

Word n + 3				[n = CIO 1,500 + (25 x unit number)]
Bit	**Name**	**Status**	**Con-trolled by**	**Unit operation**
00-01	Reserved for the system	OFF	Unit	Always zero
02	Enabled Commu-nication	OFF	Unit	CAN communications disabled, i.e. unit state <> ST5 (communicating)
		ON	Unit	CAN communications enabled, i.e. unit state = ST5 (communicating)
03	CAN message received	OFF	Unit	New configured CAN message received since last Unit Cyclic refresh
		ON	Unit	No new configured CAN message received since last Unit Cyclic refresh
04-05	Reserved for the system	---	---	Always zero
06	Send queue overflow	OFF	Unit	All messages to be sent fit in the send queue
		ON	Unit	Send queue overflow, some messages will be delayed
07	Receive queue overflow	OFF	Unit	All messages received were handled
		ON	Unit	Receive queue overflow, some received mes-sages have been discarded
08	Reserved for the system	---	---	Always zero
09	Network-Power failure	OFF	Unit	Network power OK
		ON	Unit	Error ER10 active (Power fail when communi-cating)
10	Bus off event	OFF	Unit	Bus off event did not happen since last event EV3 (enable communications)
		ON	Unit	Bus off event EV5 (bus off) has been gener-ated
11-14	Reserved for the system	---	---	Always zero
15	Error in error log	OFF	Unit	No new errors in error log since: 1. Startup 2. Last service of FINS command 2102 3. Last service of FINS command 2103
		ON	Unit	New errors in error log since: 1. Startup 2. Last service of FINS command 2102 3. Last service of FINS command 2103

4-2-4 Always zero input words (Words n+4, n+5, n+6, n+7)

Word n + 4, n + 5, n + 6, n + 7				[n = CIO 1,500 + (25 x unit number)]
Bit	**Name**	**Status**	**Con-trolled by**	**Unit operation**
00-15	Reserved for the system	---	---	Always zero

4-2-5 Number of delayed send messages (Word n+8)

When bit 06 in CIO word n+3 is true this word will show the number of delayed send messages.

Word n + 8				[n = CIO 1,500 + (25 x unit number)]
Bit	**Name**	**Status**	**Con-trolled by**	**Unit operation**
00-15	Number of delayed messages	BCD-coded	Unit	Number of messages that are delayed, see impetrated n + 3, bit 6. Range 0 - 15 (BCD)

4-2-6 Number of received messages waiting to be processed (Word n+9)

When bit 07 in CIO word n+3 is true this word will show the number of received messages waiting to be processed.

Word n + 9				[n = CIO 1,500 + (25 x unit number)]
Bit	**Name**	**Status**	**Con-trolled by**	**Unit operation**
00-15	Number of mes-saged to be pro-cessed	BCD-coded	Unit	Number of input messages. Range 0 - 15 (BCD)

4-2-7 Reserved words for the system (Words n+10 - n+24)

Word n + 10 - n + 24				[n = CIO 1,500 + (25 x unit number)]
Bit	**Name**	**Status**	**Con-trolled by**	**Unit operation**
00-15	Reserved by system	---	---	Reserved by the system

4-3 Memory areas

Memory areas

In the next paragraphs four memory areas and their relationships are described, the four memory areas are:

- send triggers area
  Every bit in this area can be a trigger to send a CAN message, located in the output buffers area. The bit is only a send trigger if a corresponding output buffer is defined for the bit, the send mode SM1 is part of the total send mode for the output buffer and the bit has a rising flank.
- receive flags area
  These area has flags indicating that a CAN message is received in the input buffer corresponding to the receive bit.
- output buffers area
  In this area one or more output buffers are present, every output buffer can have one (and only one) CAN message content for sending.
- input buffers area
  The area consists of one or more input buffers, every input buffer can have the content of one (and only one) received CAN message.

Power on

Every time after 'power on' of the unit the user needs to configure these memory areas. The memory areas are configured from within the PLC program, using FINS command 2902. The configuration steps in the following list should be done after every 'power on' of the unit, to insure correct operation:

1. Setting memory area (buffer) allocations, FINS 2902
2. Setting parameters for sending messages with 11-bit identifier, or 29-bit identifier
3. Setting parameters for receiving messages with 11-bit identifier, or 29-bit identifier
4. Enable CAN, CIO word 0 bit 4.

Note Steps 2 and 3 are optional, and their order given above is not mandatory.

4-3-1 Send Trigger Area

The location *R* of the send trigger words is configured via FINS command 2902, See "Setting the memory areas" on page 54. The words in this memory area are inputs as well as outputs. The length of the area is determined via the number of configured output message buffers. If the number of configured output message buffers is *A*, the length *L* is:

$$L = \frac{(A + 15)}{16}$$

The result is rounded down to whole numbers, remaining triggers will be ignored.

The trigger send area table has the following layout:

word *R*	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
word *R*+1	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
≀	≀	≀	≀	≀	≀	≀	≀	≀	≀	≀	≀	≀	≀	≀	≀	≀
word *R*+*L*	16**L* +15	16**L* +14	16**L* +13	16**L* +12	16**L* +11	16**L* +10	16**L* +9	16**L* +8	16**L* +7	16**L* +6	16**L* +5	16**L* +4	16**L* +3	16**L* +2	16**L* +1	16**L*

Every output message buffer has its corresponding bit, the send trigger. Both PLC and unit can be actors on a bit:

Actor	Action	Function
PLC	bit set	In mode SM1(output messages are sent by triggering a bit in trigger send area table): after the next Unit Cyclic refresh, the unit will send the corresponding message. In other modes: no function.
PLC	bit reset	Unspecified functionality: the corresponding message may sent or not.
Unit	bit set	Unspecified functionality: the unit shall not set bits in this area.
Unit	bit reset	In mode SM1(output messages are sent by triggering a bit in trigger send area table): the unit indicates that the corresponding message has been sent by clearing the send bit in the send area table. In other modes: the unit shall not reset the bit.

4-3-2 Receive Flags Area

The location *S* of the receive flags word is configured via FINS command 2902. This memory area contains input words. The length of the area is determined via the number of configured input message buffers. If the number of configured input message buffer is *B*, the length is:

$$L = \frac{(B + 15)}{16}$$

The receive flag area table has the following layout:

word *S*	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
word *S*+1	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
≀	≀	≀	≀	≀	≀	≀	≀	≀	≀	≀	≀	≀	≀	≀	≀	≀
word *S*+*L*	16**L* +15	16**L* +14	16**L* +13	16**L* +12	16**L* +11	16**L* +10	16**L* +9	16**L* +8	16**L* +7	16**L* +6	16**L* +5	16**L* +4	16**L* +3	16**L* +2	16**L* +1	16**L*

Every input message buffer has its corresponding bit. The bit value means:

ON = An input message with the configured message identifier of this input message buffer has been received since last Unit Cyclic refresh.

OFF = No input message with the configured message identifier of this input message buffer has been received since last cyclic refresh or the message has been discarded due to receive queue overflow (CIO n + 3, bit 07 is ON).

If any of these bits is set, CIO n + 3, bit 03 is ON.

if all these bits are reset, CIO n + 3, bit 03 is OFF.

4-3-3 CAN Output Message Buffers

The location *T* of the CAN output message buffers is configured via FINS command 2902. This memory area contains output words.

Given the number of configured output message buffers as *A*, the area has the layout specified below. The layout depends on the configuration supplied via the FINS commands 2903 or 2904. By selecting little of big endian format (adjust when FINS command is executed) a byte swap is performed

	Big endian:			Little endian:	
Word *T*	Data length (number of bytes)		Buffer 0	Data length (number of bytes)	
Word *T* + 1	Data byte 0	Data byte 1		Data byte 1	Data byte 0
Word *T* + 2	Data byte 2	Data byte 3		Data byte 3	Data byte 2
Word *T* + 3	Data byte 4	Data byte 5		Data byte 5	Data byte 4
Word *T* + 4	Data byte 6	Data byte 7		Data byte 7	Data byte 8
Word *T* + (1 * 5)	Data length (number of bytes)		Buffer 1	Data length (number of bytes)	
Word *T* + (1 * 5) + 1	Data byte 0	Data byte 1		Data byte 1	Data byte 0
Word *T* + (1 * 5) + 2	Data byte 2	Data byte 3		Data byte 3	Data byte 2
Word *T* + (1 * 5) + 3	Data byte 4	Data byte 5		Data byte 5	Data byte 4
Word *T* + (1 * 5) + 4	Data byte 6	Data byte 7		Data byte 7	Data byte 8
≀	≀	≀	≀	≀	≀
Word *T* + ((*A* - 1)* 5)	Data length (number of bytes)		Buffer *m*	Data length (number of bytes)	
Word *T* + ((*A* - 1)* 5) + 1	Data byte 0	Data byte 1		Data byte 1	Data byte 0
Word *T* + ((*A* - 1)* 5) + 2	Data byte 2	Data byte 3		Data byte 3	Data byte 2
Word *T* + ((*A* - 1)* 5) + 3	Data byte 4	Data byte 5		Data byte 5	Data byte 4
Word *T* + ((*A* - 1)* 5) + 4	Data byte 6	Data byte 7		Data byte 7	Data byte 8

4-3-4 CAN Input Message Buffers

The location *U* of the CAN input message buffers is configured via FINS command 2902. This memory area contains input words.

	Big endian:			Little endian:	
Word *U*	Data length (number of bytes)		Buffer 0	Data length (number of bytes)	
Word *U* + 1	Data byte 0	Data byte 1		Data byte 1	Data byte 0
Word *U* + 2	Data byte 2	Data byte 3		Data byte 3	Data byte 2
Word *U* + 3	Data byte 4	Data byte 5		Data byte 5	Data byte 4
Word *U* + 4	Data byte 6	Data byte 7		Data byte 7	Data byte 8
Word *U* + (1 * 5)	Data length (number of bytes)		Buffer 1	Data length (number of bytes)	
Word *U* + (1 * 5) + 1	Data byte 0	Data byte 1		Data byte 1	Data byte 0
Word *U* + (1 * 5) + 2	Data byte 2	Data byte 3		Data byte 3	Data byte 2
Word *U* + (1 * 5) + 3	Data byte 4	Data byte 5		Data byte 5	Data byte 4
Word *U* + (1 * 5) + 4	Data byte 6	Data byte 7		Data byte 7	Data byte 8
≀	≀	≀	≀	≀	≀
Word *U* + ((*B* - 1)* 5)	Data length (number of bytes)		Buffer *m*	Data length (number of bytes)	
Word *U* + ((*B* - 1)* 5) + 1	Data byte 0	Data byte 1		Data byte 1	Data byte 0
Word *U* + ((*B* - 1)* 5) + 2	Data byte 2	Data byte 3		Data byte 3	Data byte 2
Word *U* + ((*B* - 1)* 5) + 3	Data byte 4	Data byte 5		Data byte 5	Data byte 4
Word *U* + ((*B* - 1)* 5) + 4	Data byte 6	Data byte 7		Data byte 7	Data byte 8

Given the number of configured input message buffers as *B*, the area has the layout specified below. The layout depends on the configuration supplied via

FINS commands 2905 or 2906. By selecting little of big endian format (adjust when FINS command is executed) a byte swap is performed

SECTION 5
FINS Commands and Responses

This section describes message communications using FINS commands sent from the ladder program in the CPU Unit of the PLC, sent from the PC or any other communication unit able to send FINS commands.

5-1 Overview 48
5-2 FINS Communications 48
5-2-1 Sending/Receiving FINS Command/Responses 49
5-2-2 Units Supporting FINS Communications 50
5-2-3 FINS Command Lists 50
5-3 Using FINS Message Communications 50
5-3-1 Sending FINS Commands 51
5-3-2 Using CMND(490) 52
5-3-3 Connecting Networks for FINS Communications 53
5-4 Fins Commands and Responses 54
5-4-1 Setting the memory areas 54
5-4-2 Setting parameters for sending 11-bit ID message 57
5-4-3 Setting parameters for sending 29-bit ID message 58
5-4-4 Setting parameters for receiving 11-bit ID message 60
5-4-5 Setting parameters for receiving 29-bit ID message 62
5-4-6 Setting parameters for direct transmitting 11-bit ID message 63
5-4-7 Setting parameters for direct transmitting 29-bit ID message 64
5-4-8 Setting the CAN bit rate and sample point 66

5-1 Overview

FINS communications enable messages to be sent between nodes on a FA network when required by system conditions. The messages can be sent between PLC's, between an OMRON PLC and a master made by another company, or between slaves. They can be used to send/receive data; read time data, error histories, and other data; or control operation, e.g., by force-setting/resetting bits.

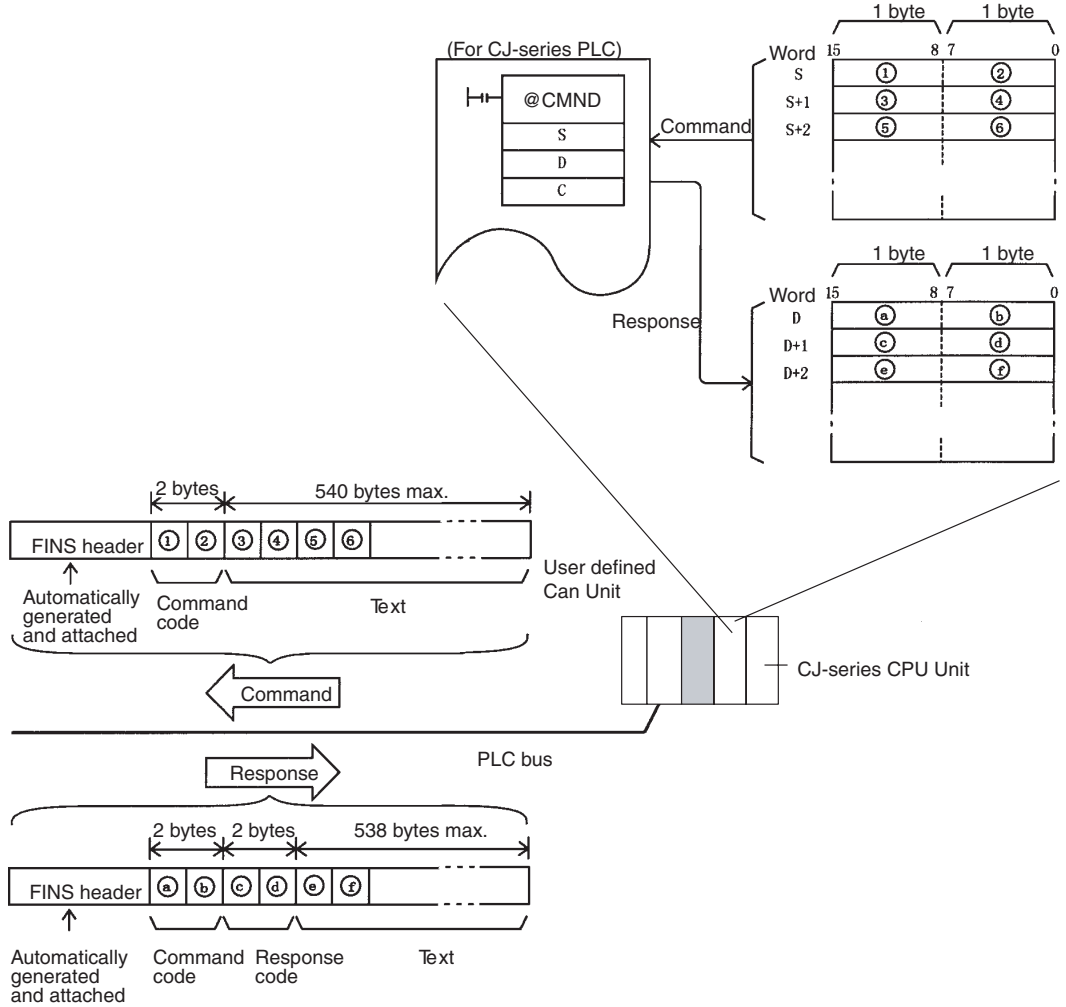
5-2 FINS Communications

The FINS communication protocol was developed by OMRON for use with factory automation control devices. FINS communications enable PLC memory read/write and operation control without extensive programming in the user program in the PLC. FINS communications use an independent system of addresses that does not rely on the addresses used in the CAN network. This enables communications not only with nodes on the CAN network, but also with devices and PLC's connected via other FA networks, such as the SYSMAC NET and SYSMAC LINK Networks.

Refer to *SYSMAC CS/CJ Series Communication Commands Reference Manual* (W342) for details on FINS commands.

5-2-1 Sending/Receiving FINS Command/Responses

FINS commands are sent using the CMND(490) instruction for CJ-series PLCs. Send/receive FINS commands and responses and the data formats used are illustrated in the following diagram. Unless otherwise specified, data is presented in hexadecimal notation.



Command Codes

Command codes are represented by a 2-byte hexadecimal code. FINS commands always begin with a 2-byte command code and any parameters that are required follow the command code.

Response Codes

Response codes are represented by a 2-byte hexadecimal code that indicates the results of command execution. The first byte provides the main response code (MRC), which classifies the results, and the second byte provides the sub-response code (SRC), which provides details on the results.

The main response codes are listed below. Refer to *SYSMAC CS/CJ Series Communication Commands Reference Manual* (W342) for further details on response codes, including sub-response codes (SRC).

Main code	Main code
00: Normal completion	20: Read not possible
01: Local node error	21: Write not possible
02: Destination node error	22: Not executable in current mode
03: Communications controller error	23: No such device

Main code	Main code
04: Unsupported setting error (service not supported)	24: Start/stop not possible
05: Routing error	25: Unit error
10: Command format error	26: Command error
11: Parameter error	30: Access right error
	40: Abort

5-2-2 Units Supporting FINS Communications

The parameters used for FINS commands depend on the Unit that is processing the command. For command details *See "Fins Commands and Responses" on page 54.*

Refer to *SYSMAC CS/CJ Series Communication Commands Reference Manual* (W342) for information on commands and responses addressed to CS/CJ-series CPU Units.

5-2-3 FINS Command Lists

Commands Addressed to User defined CAN Units

Name	Command code		Page
CONFIGURE MEMORY AREAS	29	02	54
CONFIGURE 11-BIT ID OUTPUT MESSAGE BUFFER	29	03	57
CONFIGURE 29-BIT ID OUTPUT MESSAGE BUFFER	29	04	58
CONFIGURE 11-BIT ID INPUT MESSAGE BUFFER	29	05	60
CONFIGURE 29-BIT ID INPUT MESSAGE BUFFER	29	06	62
SETTING THE CAN BIT RATE AND SAMPLE POINT	29	09	66
DIRECT TRANSMIT OF AN 11-BIT ID CAN MESSAGE	29	07	63
DIRECT TRANSMIT OF AN 29-BIT ID CAN MESSAGE	29	08	64

General Service Message

Name	Command Code		Page
ERROR LOG READ	21	02	Refer to *SYSMAC CS/CJ Series Communication Commands Reference Manual* (W342) for details on FINS commands.
ERROR LOG CLEAR	21	03	
CONTROLLER DATA READ READ FIRMWARE VERSION	05	01	

5-3 Using FINS Message Communications

FINS commands are sent from a CPU to the Unit by using the instruction CMND(490) in the PLC program.

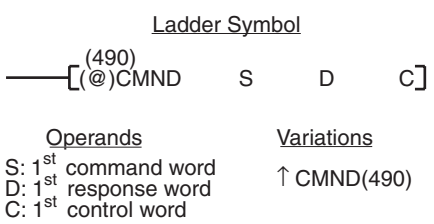
5-3-1 Sending FINS Commands

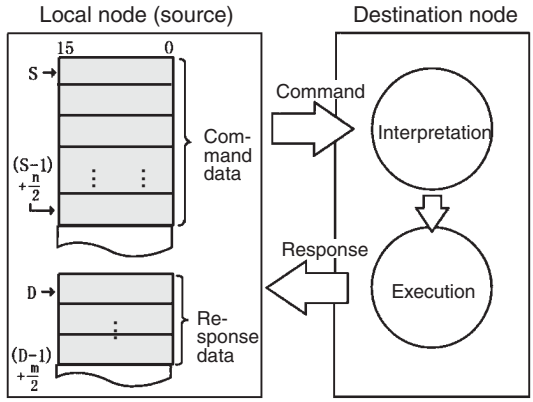
DELIVER COMMAND: CMND(490)

Description

CMND(490) can be used in the user program of the CS/CJ-series CPU Unit to send FINS commands to read/write I/O memory, read status data, change the operating mode, and perform other functions at other nodes.

CMND(490) transmits the command beginning at word S to the designated Unit at the destination node address in the designated network, and receives the response beginning at word D.





Word	Bits 00 to 07	Bits 08 to 15
C	Number of command bytes to send: 0 to 542 (i.e., #0000$_h$ to #021E$_h$) For sending a FINS message this value is the total length of the message	
C+1	Number of response bytes to receive: 0 to 542 (i.e., #0000$_h$ to #021E$_h$)	
C+2	Destination network address (0 to 127, i.e., #00$_h$ to #7F$_h$, 0 = local network)	Set to 0.
C+3	Destination unit FINS address[1]	Destination node address[2]
C+4	Bits 00 to 03: No. of retries (0 to 15, i.e., #0$_h$ to #F$_h$) Bits 04 to 07: Set to 0.	Bits 08 to 10: Transmission port number (#0$_h$ to #7$_h$) Bit 11 to 14: Set to 0. Bit 15: ON: No response. OFF: Response returned.
C+5	Response monitoring time (#0001$_h$ to #FFFF$_h$ = 0.1 to 6553.5 seconds)[3]	

Note

1. Indicates a Unit as shown in the following table.

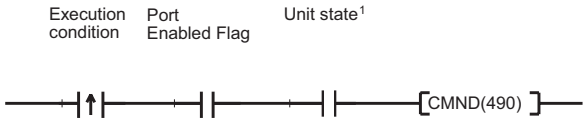
Unit	Setting
CPU Bus Unit	#10$_h$ to #1F$_h$: Unit numbers 0 to 15 #FE$_h$: The local Unit

2. Values of #00$_h$ to #FF$_h$ indicate nodes 0 to 255.
3. Designates the length of time in (0.1-s units) that the PLC retries transmission when bit 15 of C+3 is OFF and no response is received. The default value is #0000$_h$, which indicates 2 seconds. If baud rate is slow and the response monitoring time is short, an error may occur.

See *Appendix B* for an example how to use CMND(490).

5-3-2 Using CMND(490)

CMND(490) instructions normally use an AND of the Port Enabled Flag in the CPU Unit (A20200 to A2020007 corresponding to ports 0 to 7).



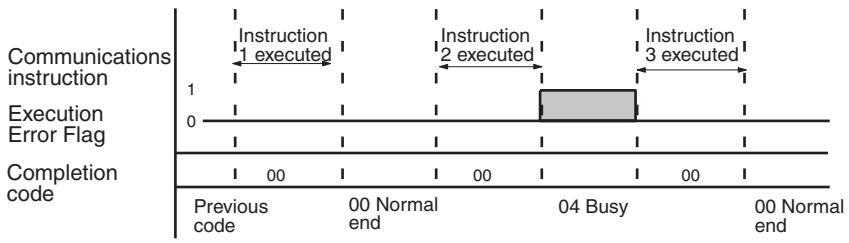
1. Unit state condition depends on the FINS command, e.g. FINS command 2907 can be sent only if the unit is in ST5 (Enabled Communication flag CIO n+3.02 is ON).

Communications Flags

Type	Name	Address		Content
		Word	Bit	
CPU Unit	Port Enabled Flag	A202	7 6 5 4 3 2 1 0 Port 7 Port 6 Port 5 Port 4 Port 3 Port 2 Port 1 Port 0	0: Execution not possible (executing) 1: Execution possible (not executing)
	Port Error Flag	A219	7 6 5 4 3 2 1 0 Port 7 Port 6 Port 5 Port 4 Port 3 Port 2 Port 1 Port 0	0: Normal completion 1: Error

Communications Flag Operation

- The Port Enabled Flag turns OFF when instructions are being sent or received (from when the instruction is issued until the response is received) and turns ON when the execution has been completed, whether normally or with an error.
- The Port Error Flag maintains its status until the next send or receive is executed.
- The Port Error Flag turns OFF when the next communications instruction is executed, regardless of whether the previous execution ended with an error.



Communications Instructions Completion Codes

The status at the completion of the execution of network communications instructions is shown at the words indicated in the following table, as FINS completion codes. During execution of the instructions the status will be 00 or

#0000_h and this will be reflected at the end of the execution of the network communications instruction.

Words	Functions
Port #0_h to #7_h Completion Codes (A203 to A210)	A203 to A210 contain the completion codes for the results of communications instruction execution for ports #0_h to #7_h, respectively.

Note For the CS/CJ-series, the completion codes are stored as 2 bytes of data (1 word) upon completion of the execution of SEND(090), RECV(098), and CMND(490). These codes are the same as the response codes for FINS commands. The first byte of the completion code is placed in bits 08 to 15 and the second byte is placed in bits 00 to 07.

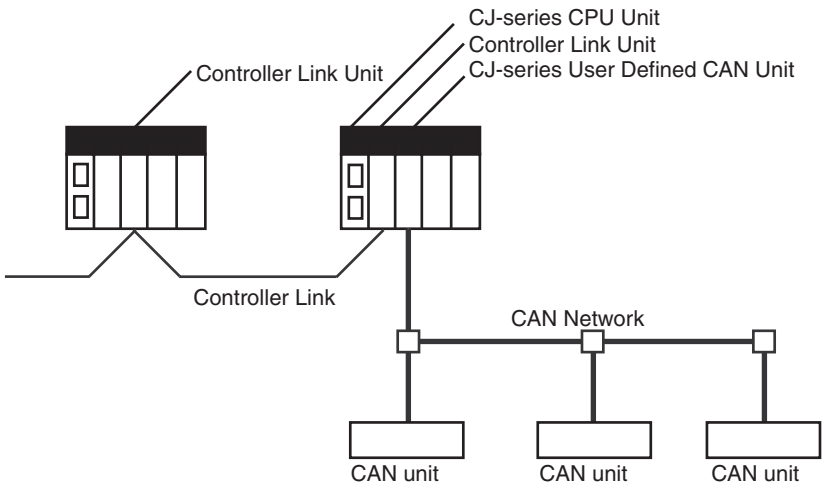
Timing of Reading Responses

Responses should be read on the rising edge (upward differentiation) of the Port Enabled Flag, as shown in the following diagram.

5-3-3 Connecting Networks for FINS Communications

CJ-series User Defined CAN Units can NOT perform FINS communications between networks. For FINS communication over networks the PLC system should be configured with other FA networks such as Controller Link or SYSMAC LINK, and the OA network Ethernet.

User Defined CAN Units can only be can be connected to the CAN networks.



When connecting networks, routing tables must be registered for the PLC CPU Units on all networks.

When connecting two or more communication units to a CJ-series CPU Unit, the Link Unit must be registered in the CJ-series PLC CPU Unit routing table (for the local network table only). Commands cannot be executed if the Link Unit is not registered.

5-4 Fins Commands and Responses

Configuration steps

In order to operate a CAN network, each unit in the network needs to be configured: either with hardware settings and or software (PLC program) configuration. The total process of network and unit configuration involves:

- setting up the physical network topology.
- setting up the bus parameters, which define the baud rate and the bus timing parameter sample point.
- defining the configuration data, i.e. defining the process data, which will be exchanged between the User Defined CAN Unit and other nodes on the CAN network.
- defining the parameterization data for the User Defined CAN Unit, which defines the filtering of message identifiers, the configuration of message buffers in the PLC system.

setting up the parameterization of the User Defined CAN Unit. Every time after power on of the unit one needs to configure memory areas and message parameterization. The software configuration steps (PLC program) that must be done after every 'power on' of the unit, to insure correct operation are:

1. Setting memory area (buffer) allocations, FINS 2902
2. Setting parameters for sending messages with 11-bit identifier, or 29-bit identifier
3. Setting parameters for receiving messages with 11-bit identifier, or 29-bit identifier
4. Enable CAN, CIO word 0 bit 4.

Note Steps 2 and 3 are optional, and their order given above is not mandatory.

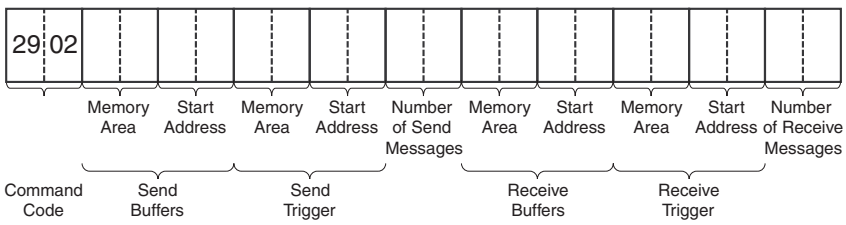
5-4-1 Setting the memory areas

FINS command 2902

With this FINS command the send buffer, send trigger, receive buffer and receive flag locations and the number of send and receive messages are set. One send or receive buffer occupies 5 words of PLC memory being the Data Length Code (DLC) and the 8 bytes of data. One send trigger or receive flag occupies one bit of PLC memory. The number of words occupied by all triggers or flags equals the number of send or receive messages divided by 16, rounded up. Trailing padding bits are ignored.

Note This command can be used only if communication has been disabled (by clearing CIO n.04) and this disabling is confirmed by the unit (by a cleared CIO n + 3.02).

Command Block



Setting	Minimum value	Maximum value
Memory Area of send buffers	#0001$_h$	#0014$_h$
Start address of send buffer	#0000$_h$	#7FFF$_h$[a]
Memory Area of send trigger	#0001$_h$	#0014$_h$
Start address of send trigger	#0000$_h$	#7FFF$_h$[1]
Number of send messages	#0000$_h$	#0280$_h$
Memory Area of receive buffers	#0001$_h$	#0014$_h$
Start address of receive buffer	#0000$_h$	#7FFF$_h$[1]
Memory Area of receive trigger	#0001$_h$	#0014$_h$
Start address of receive trigger	#0000$_h$	#7FFF$_h$[1]
Number of receive messages	#0000$_h$	#0280$_h$

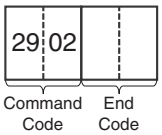
a. Actual maximum value depends on the used PLC memory area, see the next table.

The minimum and maximum values for the memory area settings in the table above, depend on the memory area. In the following table the relation between the numerical values and the type of the memory area is given, every type of memory area has it own set of minimum and maximum address values.

Memory Area	Description	Maximum address
#0001$_h$	CIO area	#17FF$_h$
#0002$_h$	DM area	#7FFF$_h$
#0003$_h$	Work area	#1FF$_h$
#0004$_h$	Holding area	#1FF$_h$
#0008$_h$	EM bank 0	#7FFF$_h$
#0009$_h$	EM bank 1	#7FFF$_h$
#000A$_h$	EM bank 2	#7FFF$_h$
#000B$_h$	EM bank 3	#7FFF$_h$
#000C$_h$	EM bank 4	#7FFF$_h$
#000D$_h$	EM bank 5	#7FFF$_h$
#000E$_h$	EM bank 6	#7FFF$_h$
#000F$_h$	EM bank 7	#7FFF$_h$
#0010$_h$	EM bank 8	#7FFF$_h$
#0011$_h$	EM bank 9	#7FFF$_h$
#0012$_h$	EM bank A	#7FFF$_h$
#0013$_h$	EM bank B	#7FFF$_h$
#0014$_h$	EM bank C	#7FFF$_h$

Note Other memory area settings will result in a error.

Response Block



End Code	Description	Condition
$\#0000_h$	Normal completion	Unit state = ST3 or ST4 and all parameters are in range
$\#1001_h$	Command length exceeds maximum command length	
$\#1002_h$	The command length is insufficient for the smallest command	
no response		Unit state = ST1 or ST2
$\#1101_h$	No area type	Unit state = ST3 or ST4 and the memory area is not available
$\#1104_h$	Address range over	Unit state = ST3 or ST4 and the end address of an area is not in range
$\#1109_h$	Mutual relation error	Unit state = ST3 or ST4 and the memory areas overlap
$\#110C_h$	Parameter error	Unit state = ST3 or ST4 and one or more parameters are not in range
$\#2201_h$	Not executable in current mode	Unit state = ST5

Note On response code 0000, event EV1 is generated. The unit configures

- Location T in *4-3-3 CAN Output Message Buffers* using the provided memory area and start address of Output Buffers
- Location R in *4-3-1 Send Trigger Area* using the provided memory area and start address of Send Triggers
- Number A in *4-3-1 Send Trigger Area* and *4-3-3 CAN Output Message Buffers* with the number of output buffers
- Location U in *4-3-3 CAN Output Message Buffers* using the provided memory area and start address of Input Buffers
- Location S in *4-3-2 Receive Flags Area* using the provided memory area and start address of Receive Flags
- Number B in *4-3-2 Receive Flags Area* and *4-3-4 CAN Input Message Buffers* with the number of input buffers

On response code 0000, the unit will set to zero

- The CAN Input Message buffers that are added to the configuration
- The Receive Flags that are added to the configuration

The Transmit Flags are not cleared.

Command message specification

Word	Function
word n	FINS command ($\#2902_h$).
word n + 1	Memory area (e.g. $\#0002_h$ for DM area)
word n + 2	Start address (e.g. $\#0000_h$ for start address DM 0)
word n + 3	Memory area (e.g. $\#0003_h$ for W area)
word n + 4	Start address (e.g. $\#0000_h$ for start address W 0)
word n + 5	Number of buffers (e.g. $\#000A_h$ for 10 buffers)
word n + 6	Memory area (e.g. $\#0002_h$ for DM area)
word n + 7	Start address (e.g. $\#0064_h$ for start address DM 100)

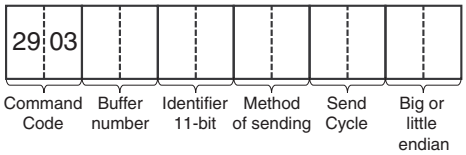
Word	Function
word n + 8	Memory area (e.g. #0003$_h$ for W area)
word n + 9	Start address (e.g. #0001$_h$ for start address W 1)
word n + 10	Number of buffers (e.g. #0005$_h$ for 5 buffers)

5-4-2 Setting parameters for sending 11-bit ID message

FINS command 2903

Per send buffer three parameters need to be set: the 29-bit identifier, the method of sending (SM1 = triggered, SM2 = on change or SM3 = cyclic), and the use of little or big endian.

Command Block



Setting	Minimum value	Maximum value
Buffer number	#0000$_h$	#027F$_h$
Identifier 11-bit	#0000$_h$	#07FF$_h$
Method of sending	#0001$_h$	#0004$_h$
Send cycle [msec]	#0000$_h$	#FFFF$_h$
Big or little endian	#0000$_h$	#0001$_h$

Note

Method of sending:

#0001$_h$	Message will be send on triggering the corresponding flag in the send trigger area.
#0002$_h$	Message will be send cyclic. The cycle time is set in milliseconds.
#0004$_h$	Message will be send on change of data.

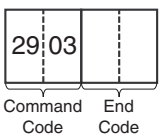
Big or little endian:

#0000$_h$	Big endian.
#0001$_h$	Little endian.

Send cycle

> 0	Value is in milliseconds.
= 0	Only valid in case the send mode is ‘on trigger’ or ‘on change’. For all other situations zero is a configuration error.

Response Block



End Code	Description	Condition
no response		Unit state = ST1 or ST2
#0000$_h$	Normal completion	Unit state = ST4 orST5 and all parameters in range
#1001$_h$	Command length exceeds maximum command length	
#1002$_h$	The command length is insufficient for the smallest command	
#1103$_h$	Address range designation error	Unit state = ST4 or ST5 and buffer number >= number of output buffers configured with command 2902
#1109$_h$	Mutual relation error	Unit state = ST4 or ST5 and send mode = 2 and send cycle = 0
#110C$_h$	Parameter error	Unit state = ST4 or ST5 and one or more parameters not in range
#2201$_h$	Not executable in current mode	Unit state = ST3
#220F$_h$	The specified service is being executed	Unit state = ST5 and send mode 'changed' and the message is being sent

Note On response code 0000, the unit configures the output message buffer identified with buffer number with the following properties:

1. Extended identifier length (11-bit)
2. Identifier number
3. Send mode, See "CAN" on page 15.
4. Send cycle time (if the send mode is 2)
5. Big or little endian format, see *4-3-3 CAN Output Message Buffers*

Command message specification

Word	Function
word n	FINS command (#2903$_h$)
word n + 1	Buffer number (e.g. #0002$_h$ for buffer 2)
word n + 2	Identifier, 11-bit (e.g. #000A$_h$ for identifier 10)
word n + 3	Method of sending (e.g. #0002$_h$ for send mode 'on cycle')
word n + 4	Send cycle (e.g. #0064$_h$ for send cycle 100 msec.)
word n + 5	Big or little endian (e.g. #0000$_h$ for big endian)

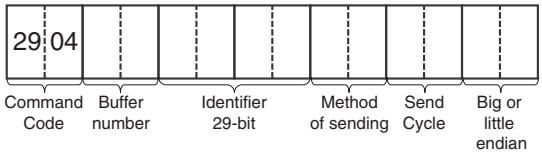
If the method of sending is cyclic the Send cycle parameter may not be set to #0000. The unit will respond with an error message. The unit of the Send cycle is in milliseconds.

5-4-3 Setting parameters for sending 29-bit ID message

FINS command 2904

Per send buffer three parameters need to be set: the 29-bit identifier, the method of sending (SM1 = triggered, SM2 = on change or SM3 = cyclic), and the use of little or big endian.

Command Block



Setting	Minimum value	Maximum value
Buffer number	#0000$_h$	#027F$_h$
Identifier 29-bit	#00000000$_h$	#1FFF FFFF$_h$
Method of sending	#0001$_h$	#0004$_h$
Send cycle	#0000$_h$	#FFFF$_h$
Big or little endian	#0000$_h$	#0001$_h$

Note

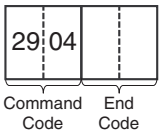
Method of sending:

#0001$_h$	Message will be send on triggering the corresponding flag in the send trigger area.
#0002$_h$	Message will be send on cyclic time. Time is set in milliseconds
#0004$_h$	Message will be send on change of data

Big or little endian:

#0000$_h$	Big endian
#0001$_h$	Little endian

Response Block



End Code	Description	Condition
no response		Unit state = ST1 or ST2
#0000$_h$	Normal completion	Unit state = ST4 orST5 and all parameters in range
#1001$_h$	Command length exceeds maximum command length	
#1002$_h$	The command length is insufficient for the smallest command	
#1103$_h$	Address range designation error	Unit state = ST4 or ST5 and buffer number >= number of output buffers configured with command 2902
#1109$_h$	Mutual relation error	Unit state = ST4 or ST5 and send mode = 2 and send cycle = 0
#110C$_h$	Parameter error	Unit state = ST4 or ST5 and one or more parameters not in range

End Code	Description	Condition
#2201$_h$	Not executable in current mode	Unit state = ST3
#220F$_h$	The specified service is being executed	Unit state = ST5 and send mode 'changed' and the message is being sent

Note On response code 0000, the unit configures the input message buffer identified with buffer number with the following properties:

1. Extended identifier length (29-bit)
2. Identifier number
3. Send mode, See "CAN" on page 15.
4. Send cycle time (if the send mode is 2)
5. Big or little endian format, see *4-3-3 CAN Output Message Buffers*

Command message specification

Word	Function
word n	FINS command (#2904$_h$)
word n + 1	Buffer number (e.g. #0000$_h$ for buffer 0)
word n + 2	MSB part identifier, 29-bit (e.g. #0000$_h$ for identifier 2, MSB + LSB)
word n + 3	LSB part identifier, 29-bit (e.g. #0002$_h$ for identifier 2, MSB + LSB)
word n + 4	Method of sending (e.g. #0001$_h$ for send mode 'on trigger')
word n + 5	Send cycle (not relevant for send mode 'on trigger')
word n + 6	Big or little endian (e.g. #0000$_h$ for big endian)

If the method of sending is cyclic the Send cycle parameter may not be set to #0000. The unit will respond with an error message. The unit of the Send cycle is in milliseconds.

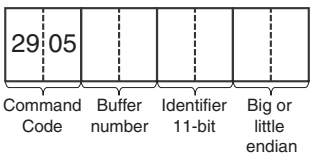
5-4-4 Setting parameters for receiving 11-bit ID message

FINS command 2905

Per receive buffer, parameters need to be set like the 11-bit identifier, the use of little or big endian.

This unit is not capable of receiving both 11-bit and 29-bit identifier messages. It will receive messages of the last configured identifier type only.

Command Block



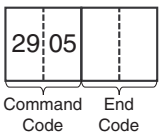
Setting	Minimum value	Maximum value
Buffer number	#0000$_h$	#027F$_h$
Identifier 11-bit	#0000$_h$	#07FF$_h$
Big or little endian	#0000$_h$	#0001$_h$

Note

Big or little endian:

$\#0000_h$	Big endian
$\#0001_h$	Little endian

Response Block



End Code	Description	Condition
no response		Unit state = ST1 or ST2
$\#0000_h$	Normal completion	Unit state = ST4 orST5 and all parameters in range
$\#1001_h$	Command length exceeds maximum command length	
$\#1002_h$	The command length is insufficient for the smallest command	
$\#1103_h$	Address range designation error	Unit state = ST4 or ST5 and buffer number >= number of input buffers configured with command 2902
$\#110C_h$	Parameter error	Unit state = ST4 or ST5 and one or more parameters not in range
$\#2201_h$	Not executable in current mode	Unit state = ST3

Note On response code 0000, the unit configures the input message buffer identified with buffer number with the following properties:

1. Extended identifier length, see restrictions below
2. Identifier number
3. Big or little endian format, see *4-3-4 CAN Input Message Buffers*

Restriction The last FINS command 2905 or 2906 sent to the unit determines whether 11-bit or 29-bit identifiers are used for all input message buffers. If the last command is 2905, 11-bit identifiers will be used. If the last command is 2906, 29-bit identifiers will be used.

Command message specification

Word	Function
word n	FINS command ($\#2905_h$)
word n + 1	Buffer number (e.g. $\#0002_h$ for buffer 2)
word n + 2	Identifier, 11-bit (e.g. $\#0002_h$ for identifier 2)
word n + 3	Big or little endian (e.g. $\#0000_h$ for big endian)

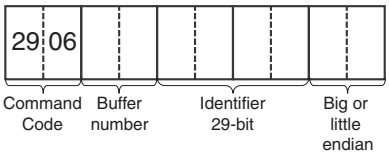
5-4-5 Setting parameters for receiving 29-bit ID message

FINS command 2906

Per receive buffer parameters need to be set like the 29-bit identifier, the use of little or big endian.

This unit is not capable of receiving both 11-bit and 29-bit identifier messages. It will receive messages of the last configured identifier type only.

Command Block



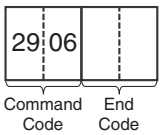
Setting	Minimum value	Maximum value
Buffer number	$\#0000_h$	$\#027F_h$
Identifier 29-bit	$\#00000000_h$	$\#1FFFFFFF_h$
Big or little endian	$\#0000_h$	$\#0001_h$

Note

Big or little endian:

$\#0000_h$	Big endian
$\#0001_h$	Little endian

Response Block



End Code	Description	Condition
no response		Unit state = ST1 or ST2
$\#0000_h$	Normal completion	Unit state = ST4 orST5 and all parameters in range
$\#1001_h$	Command length exceeds maximum command length	
$\#1002_h$	The command length is insufficient for the smallest command	
$\#1103_h$	Address range designation error	Unit state = ST4 or ST5 and buffer number >= number of input buffers configured with command 2902
$\#110C_h$	Parameter error	Unit state = ST4 or ST5 and one or more parameters not in range
$\#2201_h$	Not executable in current mode	Unit state = ST3

Note On response code 0000, the unit configures the input message buffer identified with buffer number with the following properties:

1. Extended identifier length, see restrictions below
2. Identifier number

3. Big or little endian format, see *4-3-4 CAN Input Message Buffers*

Restriction The last FINS command 2905 or 2906 sent to the unit determines whether 11-bit or 29-bit identifiers are used for all input message buffers. If the last command is 2905, 11-bit identifiers will be used. If the last command is 2906, 29-bit identifiers will be used.

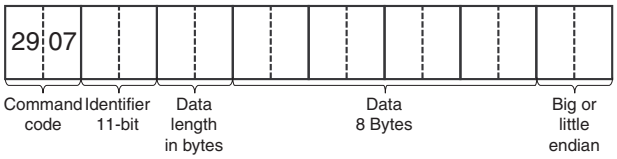
Command message specification

Word	Function
word n	FINS command (#2906$_h$)
word n + 1	Buffer number (e.g. #0002$_h$ for buffer 2)
word n + 2	MSB part identifier, 29-bit (e.g. #0000$_h$ for identifier 2, MSB + LSB)
word n + 3	LSB part identifier, 29-bit (e.g. #0002$_h$ for identifier 2, MSB + LSB)
word n + 4	Big or little endian (e.g. #0000$_h$ for big endian)

5-4-6 Setting parameters for direct transmitting 11-bit ID message

FINS command 2907 The instruction is used to transmit a user defined CAN message. This function is only for advanced users that have knowledge of the message structure of the higher layer protocol. The User Defined CAN Unit will not check the validity of the CAN message!

Command Block



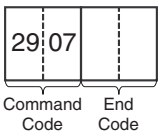
Setting	Minimum value	Maximum value
Identifier 11-bit	#0000$_h$	#07FF$_h$
Data length in bytes	#0000$_h$	#0008$_h$
Data	#00$_h$	#FF$_h$
Big or little endian	#0000$_h$	#0001$_h$

Note

Big or little endian:

#0000$_h$	Big endian
#0001$_h$	Little endian

Response Block



End Code	Description	Condition
no response		Unit state = ST1 or ST2
#0000$_h$	Normal completion	Unit state = ST5 and all parameters in range
#1001$_h$	Command length exceeds maximum command length	
#1002$_h$	The command length is insufficient for the smallest command	
#110C$_h$	Parameter error	Unit state = ST5 and one or more parameters not in range
#2201$_h$	Not executable in current mode	Unit state = ST3 or ST4
#220F$_h$	The specified service is being executed	Unit state = ST5 and send mode 'changed' and the message is being sent

Note On response code 0000, the unit sends a message with an extended identifier length with the provided identifier, data length and data. Depending on the provided endian flag.
On all response codes different from 0000, the unit will not transmit the user defined CAN message

Command message specification

Word	Function	
word n	FINS command (#2907$_h$)	
word n + 1	Identifier 11-bit (e.g. #000A$_h$ for identifier 10)	
word n + 2	Data length in bytes (e.g. #0003$_h$ first 3 bytes will be sent)	
word n + 3	Data byte 0	Data byte 1
word n + 4	Data byte 2	Data byte 3
word n + 5	Data byte 4	Data byte 5
word n + 6	Data byte 6	Data byte 7
word n + 7	Big or little endian (e.g. #0000$_h$ for big endian)[a]	

a. This example uses big endian, note that for little endian the data bytes in word n+3 to word n+6 are swapped.

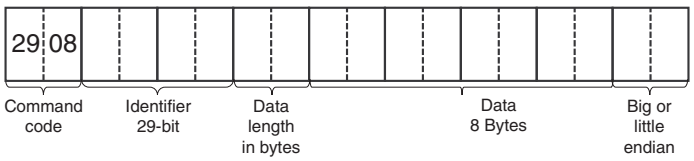
All the 8 data bytes, word n+3 until word n+6, are always part of the FINS message, the actual data length of the CAN message (word n+2) can be less than this #08$_h$. The length of the FINS command must always equal 8 words.

5-4-7 Setting parameters for direct transmitting 29-bit ID message

FINS command 2908

The instruction is used to transmit a user defined CAN message. This function is only for advanced users that have knowledge of the message structure of the higher layer protocol. The User Defined CAN Unit will not check the validity of the CAN message!

Command Block



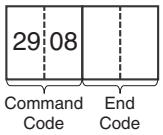
Setting	Minimum value	Maximum value
Identifier 29-bit	#00000000$_h$	#1FFFFFFF$_h$
Data length in bytes	#0000$_h$	#0008$_h$
Data	#00$_h$	#FF$_h$
Big or little endian	#0000$_h$	#0001$_h$

Note

Big or little endian:

#0000$_h$	Big endian
#0001$_h$	Little endian

Response Block



End Code	Description	Condition
no response		Unit state = ST1 or ST2
#0000$_h$	Normal completion	Unit state = ST5 and all parameters in range
#1001$_h$	Command length exceeds maximum command length	
#1002$_h$	The command length is insufficient for the smallest command	
#110C$_h$	Parameter error	Unit state = ST5 and one or more parameters not in range
#2201$_h$	Not executable in current mode	Unit state = ST3 or ST4
#220F$_h$	The specified service is being executed	Unit state = ST5 and send mode ‘changed’ and the message is being sent

Note On response code 0000, the unit sends a message with an extended identifier length with the provided identifier, data length and data. Depending on the provided endian flag.
On all response codes different from 0000, the unit will not transmit the user defined CAN message

Command message specification

Word	Function	
word n	FINS command (#2908$_h$)	
word n + 1	MSB part identifier 29-bit (e.g. #0000$_h$ for identifier 10, MSB + LSB)	
word n + 2	LSB part identifier 29-bit (e.g. #000A$_h$ for identifier 10, MSB + LSB)	
word n + 3	Data length in bytes (e.g. #0003$_h$ first 3 bytes will be sent)	
word n + 4	Data byte 0	Data byte 1
word n + 5	Data byte 2	Data byte 3

Word	Function	
word n + 6	Data byte 4	Data byte 5
word n + 7	Data byte 6	Data byte 7
word n + 8	Big or little endian (e.g. #0000$_h$ for big endian)[a]	
word n	FINS command (#2908$_h$)	

a. This example uses big endian, note that for little endian the data bytes in word n+4 to word n+7 are swapped.

All the 8 data bytes, word n+3 until word n+6, are always part of the FINS message, the actual data length of the CAN message (word n+2) can be less than this #08$_h$. The length of the FINS command must always equal 9 words.

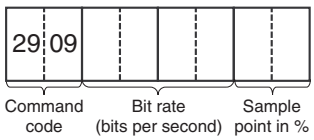
5-4-8 Setting the CAN bit rate and sample point

FINS command 2909

The bit rate can be set in 2 ways:

- The 8 bits rates recommended by CANopen can be selected with the DIP-switches on the front of the Unit. This gives the user some pre-defined bit rates.
- If the User wants to set another bit rate, the bit rate can be set via the FINS command Setting the CAN bit rate and sample point (FINS command 2909). This feature is only for experienced CAN users. If the values for these FINS settings are within the ranges specified in the table below, the bit rate is set according these settings. If a wrong setting is made, the bit rate is set to the bit rate selected with the DIP-switches at the front of the Unit. If the command is correct, the Unit will return in the response the bit rate and sample point which are in use.

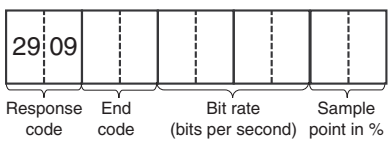
Command Block



Setting	Minimum value	Maximum value
Bit rate	#00000000$_h$	#001312D0$_h$ (1.25Mbit)
Sample point	#0000$_h$	#0064$_h$

Response Block

Apart from the standard command length checks, the following response codes:



End Code	Description	Condition
no response		Unit state = ST1 or ST2

End Code	Description	Condition
$\#0000_h$	Normal completion	Unit state = ST3 or ST4 and all parameters in range
$\#1001_h$	Command length exceeds maximum command length	
$\#1002_h$	The command length is insufficient for the smallest command	
$\#110C_h$	Parameter error	Unit state = ST3 or ST4, and one or more parameters not in range
$\#2201_h$	Not executable in current Unit state	Unit state = ST5

Note On response code 0000, the unit will set a bit rate and a sample point. Not all bit rates and all sample points in the valid range are available.

Force usage of DIP switches

Setting the bit rate and sample point to zero instructs the unit to use the DIP switch settings. The zero values causes the unit to revert to the DIP switches, any other values will result in an override of the DIP switch settings with the FINS command values.

SECTION 6
Operation

This section describes the overall specifications and the communication performance of the User Defined CAN Unit.

6-1 Performance 70
6-1-1 PLC cycle time 70
6-1-1-1 Output message evaluation time 70
6-1-1-2 Input message process time 72
6-1-2 CAN Interface. 73
6-2 Overall performance 74
6-2-1 I/O response time 74
6-2-2 Transmission of CAN messages 76
6-2-3 Reception of CAN messages 77

6-1 Performance

The overall performance of the User Defined CAN Unit depends on the performance of the host PLC interface and on the performance of the CAN interface. In the next two sub-sections, these two factors will be described first. After that, the overall performance is described.

6-1-1 PLC cycle time

PLC cycle time

The host PLC's cycle time mainly depends on the size of the PLC program and the refresh times of all units connected to its backplane(s). The size of the PLC program is application specific. Apart from optimizing the PLC program, the program execution time can only be decreased by using a faster CPU unit.

I/O refresh time

The total I/O refresh time depends on the number and types of units that are mounted on the backplane(s). Not all units refresh the same amount of data. The I/O refresh time of the User Defined CAN Unit depends on the size of the receive flags area, the size of the send trigger area, on the number of input buffers and on the number of output buffers that have been defined. All these PLC areas are defined with FINS command 2902. Refer to section 4 for details about mapping data area's on PLC memory locations and refer to section 5 for FINS commands to configure the User Defined CAN Unit, receive and send CAN messages.

6-1-1-1 Output message evaluation time

Output message evaluation time

The output message evaluation time is the time between completion if the I/O refresh of the unit and the moment an output message is set ready for sending. The duration of this period is highly user-dependent and can introduce a significant delay before the message is actually sent. The message evaluation time depends on the length and structure of the PLC-program and the send mode used for the User Defined CAN Unit. It is advised to review the timing, as the performance of the complete system (or application) is directly related to this timing. The evaluation time depends on the send mode used, and as different send modes can be combined, the total evaluation time is more or less the sum of the individual send mode evaluation times.

SM1, Triggered mode

The evaluation time in 'on trigger' mode differs with the state of the trigger, the trigger can be OFF or ON. Only when the trigger is ON, that is when the rising edge of the trigger is detected, the message in the output buffer is sent.

Every time the trigger is SET and the message is sent, the evaluation time is 10 µsec. The evaluation time is 9 µsec. every time the trigger is OFF.

The total number of output buffers is configured with FINS command 2902, FINS command 2903 and 2904 are used to associate a buffer with an identifier, a buffer with an identifier associated is called a configured buffer. The evaluation time for buffers that are not configured is 4.3 µsec.

Example 1

- Number of output buffers: 2
- Number of configured output buffers: 1

Every time the trigger of the configured message is ON, the evaluation time of all output buffers (being 2) will be: 10.0 + 4.3 = 14.3 µsec.

The evaluation time is 9.0 + 4.3 = 13.3 µsec. for the configured output buffer, in case the trigger is OFF.

Example 2

- Number of output buffers: 640
- Number of configured output buffers: 1

Every time the trigger configured message is ON, the evaluation time of all output buffers (being 640) will be: 10.0 + (4.3 * 639) = 2.76 msec.

The following guideline applies to optimizing the evaluation time for the triggered send mode:

- Set the number of output buffers with FINS command 2902 to the same amount as the number of configured output buffers (that is the total number of buffers configured with FINS command 2903 or 2904).

.

SM2, On change mode

For the On change mode, the evaluation time differs for output buffers depending on whether they contain information changes or not. In the case that data in the buffer has changed since the last evaluation, the evaluation time is 52 µsec. If the output buffer has not changed, the evaluation time depends on the length of data in all the output buffers and is in the range: 15.4 - 47.0 µsec.

The given times are for messages using the big endian format, using little endian format will increase the evaluation time with approximately 13%.

The total number of output buffers is configured with FINS command 2902, FINS command 2903 and 2904 are used to associate a buffer with an identifier, a buffer with an identifier associated is called a configured buffer. The evaluation time for buffers that are not configured is 4.3 µsec.

Example 1

- Number of output buffers: 2
- Number of configured output buffers: 1

Every time the configured message is changed, the evaluation time of all output buffers (being 2) will be: 52.0 + 4.3 = 56.3 µsec.

The evaluation time is 47.0 + 4.3 = 51.3 µsec. for output buffers that are not changed since the last evaluation, the length of the configured output buffer is assumed to be 8 characters.

Example 2

- Number of output buffers: 640
- Number of configured output buffers: 1

Every time the configured message is changed, the evaluation time of all output buffers (being 640) will be: 52.0 + (4.3 * 639) = 2.8 msec.

The following guidelines apply to optimizing the evaluation time for the on change send mode:

- Set the number of output buffers with FINS command 2902 to the same amount as the number of configured output buffers (that is the total number of buffers configured with FINS command 2903 or 2904).

- Use the big endian format in FINS commands 2903 and 2904 whenever possible, the little endian format will increase the evaluation time.

SM3, Cyclic mode

In cyclic send mode the evaluation time differs for output buffers that should be sent and for which the cyclic time has not yet elapsed. Output buffers with elapsed cyclic time have an evaluation time of 11.6 µsec. The evaluation time will be 10 µsec. as long as the cyclic time is not reached.

The total number of output buffers is configured with FINS command 2902, FINS command 2903 and 2904 are used to associate a buffer with an identifier, a buffer with an identifier associated is called a configured buffer. The evaluation time for buffers that are not configured is 4.3 µsec.

Example 1

- Number of output buffers: 2
- Number of configured output buffers: 1

Every time the configured message is sent (cyclic time elapsed), the evaluation time of all output buffers (being 2) will be: 11.6 + 4.3 = 15.9 µsec.

The evaluation time is 10.0 + 4.3 = 14.3 µsec. in case the configured output buffer is not sent.

Example 2

- Number of output buffers: 640
- Number of configured output buffers: 1

Every time the configured message is sent (cyclic time elapsed), the evaluation time of all output buffers (being 640) will be: 11.6 + (4.3 * 639) = 2.76 msec.

The following guidelines apply to optimizing the evaluation time for the cyclic send mode:

- Set the number of output buffers with FINS command 2902 to the same amount as the number of configured output buffers (that is the total number of buffers configured with FINS command 2903 or 2904).
- Sending messages cyclically instead of on change is more than 4 times faster.

Conclusions

The following conclusions apply to the evaluation times and the different send modes:

- Sending messages in the triggered mode is the fastest way to send messages, it is faster then evaluation in the on change mode and cyclic mode.
- Sending messages cyclically instead of on change is more than 4 times faster.

6-1-1-2 Input message process time

The input message process time is the time between reception of a message and the processing of that message, i.e. data is ready to be refreshed. The process time is highly user dependant and can significantly slow down the process of receiving CAN messages, in worst case messages will be lost (and missed) due to insufficient buffer capacity.

It is strongly advised to configure the input and output buffers for CAN messages according to good practice, as this configuration directly influences the performance of the application.

Process time received message

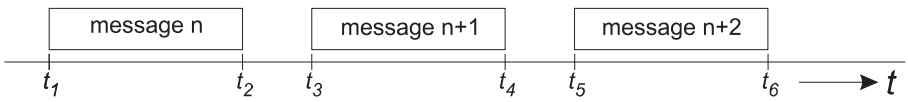
The process time of a received message is 65 µsec., only in the case that the message identifier is the one corresponding with input buffer 0. The User Defined CAN Unit compares the received identifier with the configured identifier for buffer0, next with buffer1, next with buffer 2 etc. The comparing stops if a match between the received and configured identifier is found. Therefore the process time will be 2.55 msec if the received identifier matches with the configured identifier for buffer 639. If the received identifier is not found and a total of 640 input buffers is defined, the process time of the message will be 2.50 msec.

The following guidelines apply to optimizing the process time:

1. Set the number of input buffers with FINS command 2902 to the same amount as the number of configured input buffers (that is the total number of buffers configured with FINS command 2905 or 2906).
2. Configure the order of input buffers according to the frequency of received messages, i.e. in descending order. Buffer 0 is configured with the identifier of the most frequent received message. The least frequent message should be received in the buffer with the highest number.
3. If the total number of input buffers is high and the unit will receive messages that are not configured (especially if they are very frequent): configure these message, just like they should be received by the unit. Use the same logic from the previous guideline.

6-1-2 CAN Interface

This sub-section describes the performance of the physical layer of CAN and the performance of the CAN interface of the User Defined CAN Unit. The following figure depicts the transmission of CAN messages on the bus.



$t_2 - t_1$ = *Message delay time*
$t_3 - t_2$ = *Message intermission*

Message delay time

Every message has a certain message delay time, which is mainly determined by the size of the data field in the message. CAN's bit-stuff mechanism (after every 5 consecutive equal value bits, a bit of the opposite polarity is added) can increase the message delay time by a maximum of 15%. The following formulas give the minimum and maximum message delay time.

$$\text{Message delay time} = \frac{\text{Number of bits in message}}{\text{Bit rate [bit/s]}} \quad [s]$$

$$\text{Message delay time}_{min} = \frac{44 + 8 * \text{number of data bytes}}{\text{Bit rate [bit/s]}} \quad [s]$$

$$\text{Message delay time}_{max} = \frac{\text{trunc}(47.8 + 9.6 * \text{number of data bytes})}{\text{Bit rate [bit/s]}} \quad [s]$$

(Maximum number of data bytes = 8)

6-2 Overall performance

This section describes the performance of the User Defined CAN Unit in combination with the host PLC.

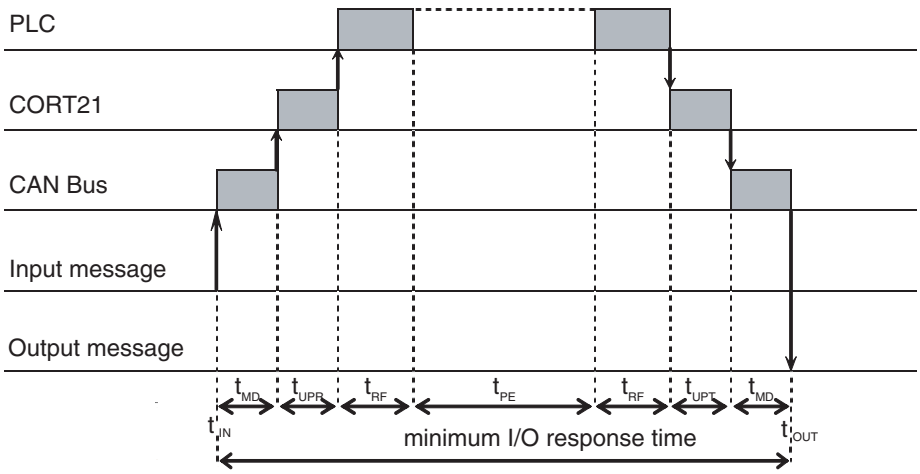
First, the bus response time is described. This is the time between an input being set (or reset) on an Input Node and an output being set (or reset) on an Output Node, all under the condition that the Input Node and Output Node are linked to the User Defined CAN Unit.

After that, the maximum transmission / reception rate of user defined CAN messages is described.

6-2-1 I/O response time

Minimum vs. maximum

The figures below show the minimum and maximum I/O response time. In the bottom figure, the processing of received messages is finished just after a PLC I/O refresh, and the received application data cannot be transferred to the PLC until the next I/O refresh. The other factors that can influence the I/O response time are for both cases identical and are explained below.

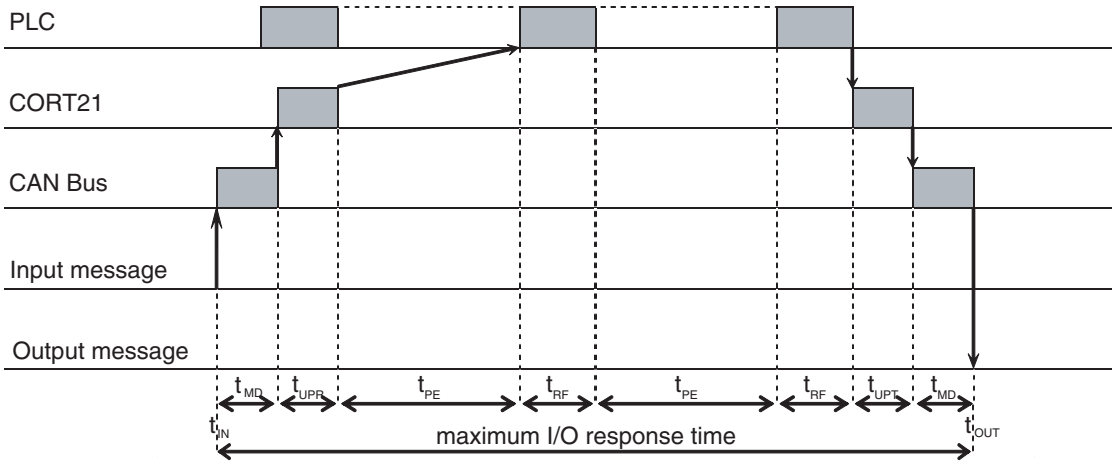


The Message Delay time

The message delay time is described in section *6-1-2 CAN Interface*.

The Unit Processing time for Transmitted / Received messages

The unit processing time for received messages depends on the message rate of all the units connected to the user Defined CAN Unit, that is the rate of received messages by the unit.
If the message rate is higher than the unit is capable of processing (see section *6-1-2 CAN Interface*), the messages will queue up causing a delay in processing. Receive queue overflow occurs when messages are received, and the queue is already full.



t_{IN} : Input Message
t_{OUT} : Output Message
t_{MD} : Message Delay time
t_{UPR} : Unit Processing time for Received messages
t_{UPT} : Unit Processing time for Transmitted messages
t_{PE} : Program Execution time
t_{RF} : I/O Refresh time

The unit processing time of messages to be transmitted depends on the number of send triggers and the (send) mode of each output buffer. If many send triggers change at (or near) the same time, or if many output buffers are changed at the same time, or have the same time triggering, the transmission of output buffers will queue up causing a delay in transmission. If the rate at which the output buffers change state requires a higher transmission rate than the unit is capable of (see section *6-1-2 CAN Interface*), a transmit queue overflow occurs.

Transmit queue overflow errors can be solved by reducing the rate at which the output buffers change state and therefore requests for transmission are generated.

Receive queue overflow errors can be solved by decreasing the PLC cycle time or by reducing the rate at which the other units transmit messages.

In case of high bus loads (over 50%), additional delays can be caused by high-priority (low identifier) messages which delay the transmission of lower priority messages.

The I/O Refresh time

Refer to section *6-1-1 PLC cycle time* for the I/O refresh time of the User Defined CAN Unit.

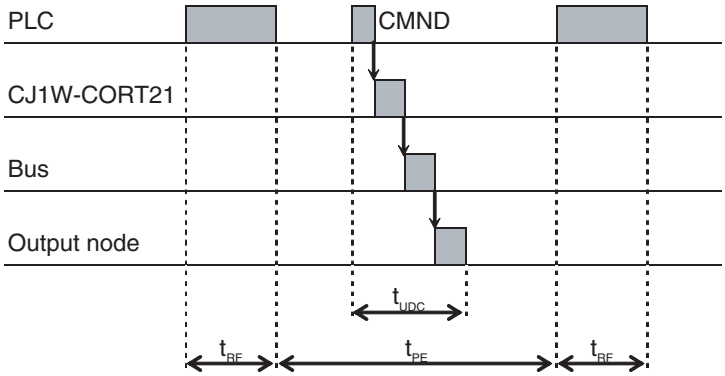
The Program Execution time

The Program Execution time depends on the number and type of PLC instructions in the ladder program. Refer to the PLC Operation Manual for the execution times of the PLC instructions.

6-2-2 Transmission of CAN messages

Direct transmitting

The following figure shows the sequence of direct transmitting a CAN message, using FINS command 2907 or 2908.



t_{UDC} : The transmission time of a user defined CAN message

t_{RF} : The I/O Refresh time

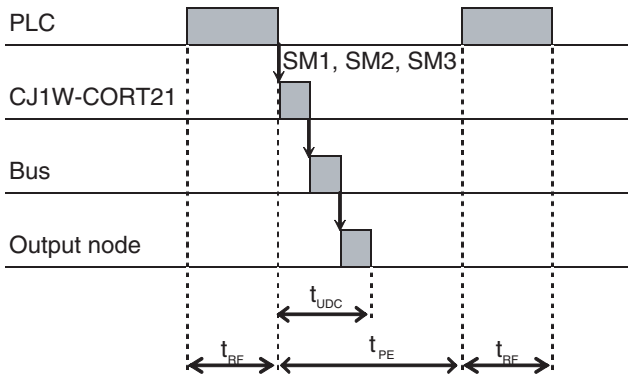
t_{PE} : The Program Execution time

Refer to the PLC Operation manual for the execution time of a CMND PLC instruction. After the content of the CMND is transferred to the User Defined CAN Unit, the unit immediately relays the contained message to the CAN transmission interface. The actual transmission delay of the message depends on the bus load and the message priority (identifier). Refer to section *6-1-2 CAN Interface* for the message delay time on the bus. The time required for an output to be set or a value in the Output Node to be changed depends on the characteristics of the Output node.

The minimum time between the execution of the CMND instruction in the PLC program and the message being transmitted on the bus is 0.5 ms. Depending on the configuration (FINS command 2902) this time can be sufficient longer, up to a maximum of 20 msec.

Transmitting with send mode

Sending messages with SM1, SM2 or SM3 differs from direct sending CAN messages: there is no delay between the start of 'program execution' and passing the message to the User Defined CAN Unit.



t_{UDC} : The transmission time of a user defined CAN message

t_{RF} : The I/O Refresh time

t_{PE} : The Program Execution time

6-2-3 Reception of CAN messages

In order to receive CAN messages, an CMND instruction (FINS command) must be executed to configure the unit regarding:

- which PLC memory locations to copy the received messages to.
- which messages to receive (the identifier defined for an input buffer).

This only needs to be done once at start-up. Immediately after the execution of the CMND instruction(s), FINS command(s), the unit is ready to receive messages.

The User Defined CAN Unit can store up to 15 received messages in its internal buffer, the receive queue. These messages in the receive queue are filtered (according to the configuration of the input buffers) and transferred to the PLC. It may take several PLC cycles, to filter and transfer a CAN message to the PLC.

The performance of the unit depends strongly on the number of input- and output buffers. If a lot of input- and/or output buffers are used, the unit needs time to process all them. Try to limit the number of input and/or output buffers. Also the PLC cycle time influences the performance. If the PLC cycle time is long, the receive queue in the unit may overflow. Try to limit the PLC cycle time.

SECTION 7
Troubleshooting and Maintenance

This section describes error processing, troubleshooting procedures and periodic maintenance operations, needed to keep the CAN network operating properly. We recommend reading through the error processing procedures before operation so that operating errors can be identified and corrected more quickly.

7-1 Troubleshooting the User Defined CAN Unit 80
7-2 Indicators 81
7-2-1 Run Led 82
7-2-2 Err Led 82
7-2-3 Two 7-segment display 83
7-2-4 Two dot indicators 83
7-3 Error Log Functions 84
7-3-1 Error Log Table 84
7-3-2 Error Codes and Detail Codes 84
7-4 Error responses from FINS commands 85
7-4-1 FINS command 2902 Configure memory area 85
7-4-2 FINS command 2903 Configure 11-bit ID output message buffer . . 86
7-4-3 FINS command 2904 Configure 29-bit ID output message buffer . . 87
7-4-4 FINS command 2905 Configure 11-bit ID input message buffer . . . 88
7-4-5 FINS command 2906 Configure 29-bit ID input message buffer . . . 88
7-4-6 FINS command 2907 direct transmit of an 11-bit ID CAN message 89
7-4-7 FINS command 2908 direct transmit of an 29-bit ID CAN message 90
7-4-8 FINS command 2909 setting the CAN bit rate and sample point . . 90
7-5 Using status information 91
7-6 Maintenance and Replacement 93
7-6-1 Cleaning 93
7-6-2 Inspection 93
7-6-3 Replacing Faulty Units 94

7-1 Troubleshooting the User Defined CAN Unit

The User Defined CAN Unit uses several error detection and error handling mechanisms.

User Defined CAN is based on the serial bus protocol of CAN. The data link layer of the CAN protocol combines 5 error detection mechanisms (CRC check, frame check, Ack check, bit check, bit stuffing check). This combination results in a Hamming distance of 6. This means that at least 6 bits in the message frame must have been disturbed to possibly remain undetected. The overall residual error probability is extremely low and this makes CAN-based protocols very reliable and suitable for harsh environments.

The User Defined CAN Unit also has some manufacturer specific status indication mechanisms

Status words CIO word n + 3 indicates the current state of the User Defined CAN Unit.

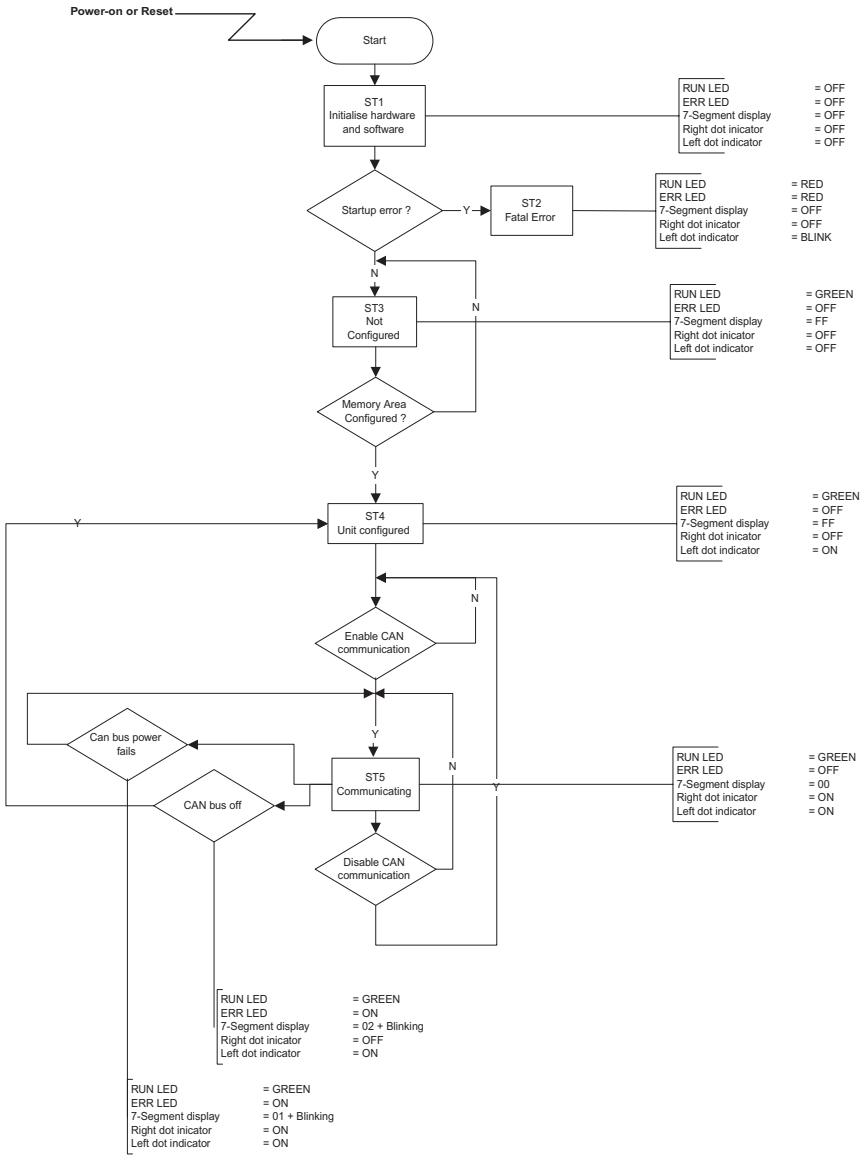
LEDs, 7-segment display Visual status indication of the User Defined CAN Unit

Error log Some errors are logged in the error log, these errors can even be read after a restart.

Error handling This chapter will give a detailed description of these higher layer error handling and status mechanisms.

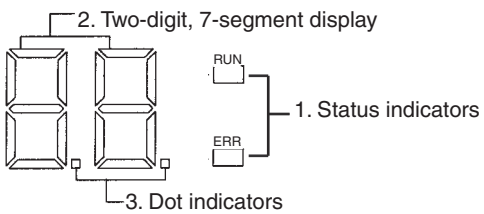
Flow diagrams to get a better understanding of how the error handling and status mechanisms are related and this chapter concludes with some flow diagrams that describe the basic functionality of the Unit.

Startup sequence In the following diagram the startup sequence is shown, including the status of all indicators of the unit.



7-2 Indicators

The User Defined CAN Unit has several indicators on front. Two 7 segment displays with dots and two error leds (see picture below).



7-2-1 Run Led

Run indicator			
Color	**Status**	**Condition**	**Possible remedy**
Green	Off	No power or in ST1 or ST2	Make sure that the unit is wired and mounted correctly, switch the power on or wait for until unit as initialized.
	On	State is ST3, ST4 or ST5	Configure the unit with FINS command 2902 or enable communication by setting CIO n.04. Enabled communication is confirmed when CIO n + 3.02 is ON or the unit is waiting for CAN messages.
Red	Off	No power or in ST1 or ST2	Make sure that the unit is wired and mounted correctly, switch the power on or wait for until unit as initialized.
	Blink	ST1 (PC21 Startup error) or ST2 (Hardware error)	Create I/O table or change the unit number. If problem persists, contact distributer.
	On	Fatal error in Unit.	Restart Unit. If problem persists, contact distributer.

7-2-2 Err Led

Err indicator			
Color	**Status**	**Condition**	**Possible remedy**
Red	Off	No error active	The unit has no active error and diagnostic functions do not detect any error condition.
	On	One of more of the following errors are active, • Hardware error • No CAN configuration • CAN network power fail, Unit is Bus off • Fatal error in Unit • CPU Watchdog Time-out • PC21 bus error • Cyclic refresh time-out • I/O table error	• Make sure that the unit is wired and mounted correctly, switch the power on or wait until the unit as initialized. • Configure the Unit. • Correct the CAN network, restore the power on the network. • Create the PLC I/O table. • Make sure the Cyclic refresh is enabled • Restart the Unit If problem persists, contact distributer.

7-2-3 Two 7-segment display

Indicator two 7-segment display			
Color	**Status**	**Condition**	**Possible remedy**
Red	All off	No power or in ST1 or ST2	Make sure that the unit is wired and mounted correctly, switch the power on or wait for unit to initialize. Configure the User Defined CAN Unit with FINS commands.
	Displays: FF	State is ST3 or ST4	If unit is not configured then configure the unit with FINS command 2902. When unit already has a configuration, enable communication by setting CIO n.04. This enabling is confirmed when CIO n + 3.02 is set.
	Displays:00	State is ST5	CAN messages can be sent and received
	Blinks: 01	CAN Bus power fail	Restore power on the CAN bus
	Blinks: 02	CAN Bus off	Check the CAN network
	Displays: H1	ST1 (PC21 Startup error)	Create I/O table or change the unit number.
	Displays: EC	ST1 (PC21 Startup error)	Create I/O table or change the unit number.
	Blinks: EC	ER2	Create I/O table, change the unit number.

7-2-4 Two dot indicators

Two dot indicators				
Indicator	**Color**	**Status**	**Condition**	**Possible remedy**
Left dot	Red	On	Unit is configured, and communication is enabled or disabled, state is ST4 or ST5	Enable communication by setting CIO n.04 This is confirmed when CIO n + 3.02 is ON or communication is enabled. CAN message can be sent and received
		Blink	Unit state is ST2, Initialization error.	Make sure that the unit is wired and mounted correctly, switch the power on or wait for unit to initialize. Configure the unit with FINS command 2902
		Off	Unit state is ST1, ST2 or ST3, unit has a startup error or is not (yet) configured	Make sure that the unit is wired and mounted correctly, switch the power on or wait for unit to initialize. Configure the unit with FINS command 2902
Right dot	Red	On	Unit state is ST5	Communication is enabled
		Off	Communication is disabled, or unit is in state ST1, ST2 or ST3 or ST4	Enable communication by setting CIO n.04. This is confirmed when CIO n + 3.02 is ON. Make sure that the unit is wired and mounted correctly, switch the power on or wait for until the unit has initialized. Configure the unit with FINS command 2902

7-3 Error Log Functions

Errors detected by the User Defined CAN Unit are stored in the error log along with the date and time of their occurrence. The error log can be read, cleared, and monitored using FINS commands.

7-3-1 Error Log Table

Error Log Table

Each time an error occurs, one error record is recorded in the User Defined CAN Unit's RAM error log table. The table can record up to 15 records. If another error occurs when the table is full, the oldest record will be erased to make room for the new error record.

The error log table records the following information.

- Error code
- Detail code
- Time of occurrence (The CPU Unit's time is used for the time stamp.)

Error Log Storage Area

When an error is detected, information on the error and the time stamp are stored in the Unit's internal RAM as an error log record. Serious errors are recorded in EEPROM as well as RAM. The error log records in EEPROM are retained even when the Unit's power is turned OFF or the Unit is restarted. The error log records in EEPROM are copied to RAM when the User Defined CAN Unit is powered-up.

When the error log is read with a FINS command, only the error log records in RAM are read. When the error log is cleared with a FINS command, the error log records in RAM and EEPROM are erased.

Reading and Clearing the Error Log

The error log table can be read or cleared by sending a FINS command to the User Defined CAN Unit. Use the User Defined CAN Unit's unit address as the FINS command's destination unit address. (The unit address is the unit number+10 Hex.).

FINS command 2102 is used to read the error history, issuing FINS command 2103 will clear the error log. Refer to the *SYSMAC CS/CJ Series Communication Commands Reference Manual* (W342) for details on using these FINS commands.

Time Information

The CPU Unit's time information is used for the time stamps in the User Defined CAN Unit's error log records. If the time information cannot be read from the CPU Unit, the time stamp will contain all zeroes.

The User Defined CAN Unit's time stamps will not be correct or consistent unless the time has been set correctly in the CPU Unit. Moreover, if the battery is replaced in a CS/CJ-series CPU Unit, the time must be set again the next time that power is turned ON.

7-3-2 Error Codes and Detail Codes

Error code (Hex)	ID	Error	Detail code		Record stored in EEPROM
			First byte	Second byte	
#0001$_h$	ER1	CPU Unit watchdog timer error	#00$_h$	#00$_h$	Yes
#0002$_h$	ER2	CPU Unit service monitoring error (The servicing time from the CPU Unit is not fixed.)	Monitoring time (ms)		Yes
#000E$_h$	ER3	PC21 Bus error	#00$_h$	#00$_h$	Yes

Error code (Hex)	ID	Error	Detail code		Record stored in EEPROM
			First byte	Second byte	
#0601$_h$	ER4	CPU Bus Unit Error	Undefined Contents		Yes
#0602$_h$	ER5	Error log read error	#01$_h$	#06$_h$	Yes
#0602$_h$	ER6	Error log write error	#02$_h$	#06$_h$	No
#0602$_h$	ER7	Network parameter read error	#01$_h$	#02$_h$	Yes
#0602$_h$	ER8	Network parameter write error	#02$_h$	#02$_h$	No
#0201$_h$	ER9	Network parameter file lost	#00$_h$	#00$_h$	No
#0340$_h$	ER10	Network power fail	#00$_h$	#02$_h$	Yes

Note The error information is not written to EEPROM when a memory error occurs in the error log area (EEPROM).

7-4 Error responses from FINS commands

In this chapter all error responses of the useful FINS-commands of the User Defined CAN Unit will be discussed.

7-4-1 FINS command 2902 Configure memory area

Error code	Description	Condition	Possible remedy
no response		Unit state = ST1 or ST2	Restart the PLC, make sure that the unit is mounted and wired correctly
#0000$_h$	Normal completion	Unit state = ST3 or ST4 and all parameters are in range	Everything is OK. Command is executed.
#1001$_h$	Command length exceeds maximum command length		Too many parameters sent in the command. Correct the command and resend.
#1002$_h$	The command length is insufficient for the smallest command		Too few parameters sent in the command. Correct the command and resend.
#1101$_h$	No area type	Unit state = ST3 or ST4 and the memory area is not available	Filled in memory area type is not valid, please fill in a correct memory area type #0001$_h$ - #0004$_h$ or #0008$_h$ - #0014$_h$, See “Setting the memory areas” on page 54.
#1104$_h$	Address range exceeded	Unit state = ST3 or ST4 and the end address of an area is not in range	Correct the memory area size
#1109$_h$	Mutual relation error	Unit state = ST3 or ST4 and the memory areas overlap	Memory area overlap, fill in a free memory area. See “CIO Area Words” on page 37. for minimum memory size

Error code	Description	Condition	Possible remedy
#110C$_h$	Parameter error	Unit state = ST3 or ST4 and one or more parameters are not in range	Correct the parameters, the number of send and receive message may not exceed a number of #0280$_h$
#2201$_h$	Not executable in current mode	Unit state = ST5	This FINS command can only be executed if communication is disabled (by clearing CIO 0.04 and this disabling is confirmed when CIO 3.2 is false)

Note For further information see *Setting the memory areas on page 54*

7-4-2 FINS command 2903 Configure 11-bit ID output message buffer

Error code	Description	Condition	Possible remedy
no response		Unit state = ST1 or ST2	Restart the PLC, make sure that the unit is mounted and wired correctly
#0000$_h$	Normal completion	Unit state = ST4 orST5 and all parameters in range	Everything is OK. Command is executed.
#1001$_h$	Command length exceeds maximum command length		Too many parameters sent in the command. Correct the command and resend.
#1002$_h$	The command length is insufficient for the smallest command		Too few parameters sent in the command. Correct the command and resend.
#1103$_h$	Address range designation error	Unit state = ST4 or ST5 and buffer number >= number of output buffers configured with command 2902	Number of output buffers exceeds the number of output buffers configured by FINS command 2902. Increase the number of output buffers with FINS command 2902 or select a output buffer, smaller than de maximum of output buffers configured by FINS command 2902.
#1109$_h$	Mutual relation error	Unit state = ST4 or ST5 and send mode = 2 and send cycle = 0	Set the send cycle different from zero, or change the send mode.
#110C$_h$	Parameter error	Unit state = ST4 or ST5 and one or more parameters not in range	Correct the parameters, See “Setting parameters for sending 11-bit ID message” on page 57.
#2201$_h$	Not executable in current mode	Unit state = ST3	Nothing configured, please perform FINS command 2902 before executing this command
#220F$_h$	The specified service is being executed	Unit state = ST5 and send mode changed and CAN message currently being sent	Disable communications, reconfigure the buffer and enable communnications.

Note For further information see *Setting parameters for sending 11-bit ID message on page 57*

7-4-3 FINS command 2904 Configure 29-bit ID output message buffer

Error code	Description	Condition	Possible remedy
no response		Unit state = ST1 or ST2	Restart the PLC, make sure that the unit is mounted and wired correctly
#0000$_h$	Normal completion	Unit state = ST4 orST5 and all parameters in range	Everything is OK. Command is executed.
#1001$_h$	Command length exceeds maximum command length		Too many parameters sent in the command. Correct the command and re-send.
#1002$_h$	The command length is insufficient for the smallest command		Too few parameters sent in the command. Correct the command and re-send.
#1103$_h$	Address range designation error	Unit state = ST4 or ST5 and buffer number >= number of output buffers configured with command 2902	Number of output buffers exceeds the number of output buffers configured by FINS command 2902. Increase the number of output buffers with FINS command 2902 or select a output buffer, smaller than de maximum of output buffers configured by FINS command 2902.
#1109$_h$	Mutual relation error	Unit state = ST4 or ST5 and send mode = 2 and send cycle = 0	Set the send cycle different from zero, or change the send mode.
#110C$_h$	Parameter error	Unit state = ST4 or ST5 and one or more parameters not in range	Correct the parameters, See “Setting parameters for sending 29-bit ID message” on page 58.
#2201$_h$	Not executable in current mode	Unit state = ST3	Nothing configured, please perform FINS command 2902 before executing this command
#220F$_h$	The specified service is being executed	Unit state = ST5 and send mode changed and CAN message currently being sent	Disable communications, reconfigure the buffer and enable communnications.

Note For further information see *Setting parameters for sending 29-bit ID message on page 58*

7-4-4 FINS command 2905 Configure 11-bit ID input message buffer

Error code	Description	Condition	Possible remedy
no response		Unit state = ST1 or ST2	Restart the PLC, make sure that the unit is mounted and wired correctly
#0000$_h$	Normal completion	Unit state = ST4 orST5 and all parameters in range	Everything is OK. Command is executed.
#1001$_h$	Command length exceeds maximum command length		Too many parameters sent in the command. Correct the command and resend.
#1002$_h$	The command length is insufficient for the smallest command		Too few parameters sent in the command. Correct the command and resend.
#1103$_h$	Address range designation error	Unit state = ST4 or ST5 and buffer number >= number of input buffers configured with command 2902	Number of input buffers exceeds the number of input buffers configured by FINS command 2902. Increase the number of input buffers with FINS command 2902 or select a input buffer, smaller than de maximum of input buffers configured by FINS command 2902.
#110C$_h$	Parameter error	Unit state = ST4 or ST5 and one or more parameters not in range	Correct the parameters, See “Setting parameters for receiving 11-bit ID message” on page 60.
#2201$_h$	Not executable in current mode	Unit state = ST3	Nothing configured, please perform FINS command 2902 before executing this command

Note For further information see *Setting parameters for receiving 11-bit ID message on page 60*

7-4-5 FINS command 2906 Configure 29-bit ID input message buffer

Error code	Description	Condition	Possible remedy
no response		Unit state = ST1 or ST2	Restart the PLC, make sure that the unit is mounted and wired correctly
#0000$_h$	Normal completion	Unit state = ST4 orST5 and all parameters in range	Everything is OK. Command is executed.
#1001$_h$	Command length exceeds maximum command length		Too many parameters sent in the command. Correct the command and resend.
#1002$_h$	The command length is insufficient for the smallest command		Too few parameters sent in the command. Correct the command and resend.

Error code	Description	Condition	Possible remedy
#1103$_h$	Address range designation error	Unit state = ST4 or ST5 and buffer number >= number of input buffers configured with command 2902	Number of input buffers exceeds the number of input buffers configured by FINS command 2902. Increase the number of input buffers with FINS command 2902 or select a input buffer, smaller than de maximum of input buffers configured by FINS command 2902.
#110C$_h$	Parameter error	Unit state = ST4 or ST5 and one or more parameters not in range	Correct the parameters, See "Setting parameters for receiving 29-bit ID message" on page 62.
#2201$_h$	Not executable in current mode	Unit state = ST3	Nothing configured, please perform FINS command 2902 before executing this command

Note For further information see *Setting parameters for receiving 29-bit ID message on page 62*

7-4-6 FINS command 2907 direct transmit of an 11-bit ID CAN message

Error code	Description	Condition	Possible remedy
no response		Unit state = ST1 or ST2	Restart the PLC, make sure that the unit is mounted and wired correctly
#0000$_h$	Normal completion	Unit state = ST5 and all parameters in range	Everything is OK. Command is executed.
#1001$_h$	Command length exceeds maximum command length		Too many parameters sent in the command. Correct the command and resend.
#1002$_h$	The command length is insufficient for the smallest command		Too few parameters sent in the command. Correct the command and resend.
#110C$_h$	Parameter error	Unit state = ST5 and one or more parameters not in range	Correct the parameters,
#2201$_h$	Not executable in current mode	Unit state = ST3 or ST4	Nothing configured, please perform FINS command 2902 before executing this command and enable communications by setting CIO n.04 ON (confirmed with CIO n+3.02 ON).
#220F$_h$	The specified service is being executed	Unit state = ST5 and CAN message currently being sent	Send the message less frequent.

Note For further information see *Setting parameters for direct transmitting 11-bit ID message on page 63*

7-4-7 FINS command 2908 direct transmit of an 29-bit ID CAN message

Error code	Description	Condition	Possible remedy
no response		Unit state = ST1 or ST2	Restart the PLC, make sure that the unit is mounted and wired correctly
#0000$_h$	Normal completion	Unit state = ST5 and all parameters in range	Everything is OK. Command is executed.
#1001$_h$	Command length exceeds maximum command length		Too many parameters sent in the command. Correct the command and resend.
#1002$_h$	The command length is insufficient for the smallest command		Too few parameters sent in the command. Correct the command and resend.
#110C$_h$	Parameter error	Unit state = ST5 and one or more parameters not in range	Correct the parameters,
#2201$_h$	Not executable in current mode	Unit state = ST3 or ST4	Nothing configured, please perform FINS command 2902 before executing this command and enable communications by setting CIO n.04 ON (confirmed with CIO n+3.02 ON).
#220F$_h$	The specified service is being executed	Unit state = ST5 and CAN message currently being sent	Send the message less frequent.
Error code	**Description**	**Condition**	**Possible remedy**
no response		Unit state = ST1 or ST2	Restart the PLC, make sure that the unit is mounted and wired correctly

Note For further information see *Setting parameters for direct transmitting 29-bit ID message on page 64*

7-4-8 FINS command 2909 setting the CAN bit rate and sample point

Error code	Description	Condition	Possible remedy
no response		Unit state = ST1 or ST2	Restart the PLC, make sure that the unit is mounted and wired correctly
#0000$_h$	Normal completion	Unit state = ST3 orST4 and all parameters in range	Everything is OK. Command is executed.
#1001$_h$	Command length exceeds maximum command length		Too many parameters sent in the command. Correct the command and resend.

Error code	Description	Condition	Possible remedy
$\#1002_h$	The command length is insufficient for the smallest command		Too few parameters sent in the command. Correct the command and resend.
$\#110C_h$	Parameter error	Unit state = ST4 or ST5 and one or more parameters not in range	Correct the parameters, See "Setting the CAN bit rate and sample point" on page 66.
$\#2201_h$	Not executable in current mode	Unit state = ST5	This FINS command can only be executed if communication is disabled (by clearing CIO 0.04 and this disabling is confirmed when CIO 3.2 is false)

Note For further information see *Setting the CAN bit rate and sample point on page 66*

7-5 Using status information

The Unit status can be read by analyzing the CIO-words. In the next table there is an overview of all information which can be analyzed. The Unit uses only CIO word n +3, n + 8 and n + 9 for status information.

Word n + 3				[n = CIO 1,500 + (25 x unit number)]
Bit	**Name**	**Status**	**Controlled by**	**Unit operation**
02	Enabled Communication	OFF	Unit	CAN communications disabled, i.e. unit state <> ST5 (communicating)
		ON	Unit	CAN communications enabled, i.e. unit state = ST5 (communicating)
03	CAN message received	OFF	Unit	New configured CAN message received since last cyclic refresh
		ON	Unit	No new configured CAN message received since last cyclic refresh
06	Send overflow	OFF	Unit	All messages to be sent fit in the send queue
		ON	Unit	Send queue overflow, some messages will be delayed
07	Receive overflow	OFF	Unit	Receive queue overflow, some received messages have been discarded
		ON	Unit	All messages received were handled
09	Network-Power failure	OFF	Unit	Network power OK
		ON	Unit	Error ER10 active (Power fail when communicating)
10	Bus off event	OFF	Unit	Bus off event did not happen since last event EV3 (enable communications)
		ON	Unit	Bus off event EV5 (bus off) has been generated
15	Error in error log	OFF	Unit	No new errors in error log since: 1. Startup 2. Last service of FINS command 2102 3. Last service of FINS command 2103
		ON	Unit	New errors in error log since: 1. Startup 2. Last service of FINS command 2102 3. Last service of FINS command 2103

Word n + 8				[n = CIO 1,500 + (25 x unit number)]
Bit	**Name**	**Status**	**Con-trolled by**	**Unit operation**
00-15	Number of delayed messages	BCD-coded	Unit	Number of messages that are delayed, see send overflow, n + 3, bit 6. Range 0 - 15 (BCD)

Word n + 9				[n = CIO 1,500 + (25 x unit number)]
Bit	**Name**	**Status**	**Con-trolled by**	**Unit operation**
00-15	Number of mes-saged to be pro-cessed	BCD-coded	Unit	Number of input messages. Range 0 - 15 (BCD)

7-6 Maintenance and Replacement

This section describes the routine cleaning and inspection recommended as regular maintenance as well as the Unit replacement procedure required if the User Defined CAN Unit fails.

7-6-1 Cleaning

Clean the User Defined CAN Units regularly as described below in order to keep the network in its optimal operating condition.

- Wipe the Unit daily with a dry, soft cloth.
- When a spot can't be removed with a dry cloth, dampen the cloth with a neutral cleanser (2% solution), wring out the cloth, and wipe the Unit.
- A smudge may remain on the Unit from gum, vinyl, or tape that was left on for a long time. Remove the smudge when cleaning.

⚠ **Caution** Never use volatile solvents such as paint thinner, benzene, or chemical wipes. These substances could damage the surface of the Unit.

7-6-2 Inspection

Be sure to inspect the system periodically to keep it in its optimal operating condition. In general, inspect the system once every 6 to 12 months, but inspect more frequently if the system is used with high temperature or humidity or under dirty/dusty conditions.

Inspection Equipment

Prepare the following equipment before inspecting the system.

Required equipment:

Have a standard and Phillips-head screwdriver, multimeter, alcohol, and a clean cloth.

Equipment required occasionally:

Depending on the system conditions, a synchroscope, oscilloscope, thermometer, or hygrometer (to measure humidity) might be needed.

Inspection Procedure

Check the items in the following table and correct any items that are below standard.

Item		Standard	Equipment
Environmental conditions	Ambient and cabinet temperature	0° C to 55° C	Thermometer
	Ambient and cabinet humidity (with no condensation or icing)	10% to 90%	Hygrometer
	Dust/dirt accumulation	None	---
Installation	Are the Units installed securely?	No looseness	---
	Are the communications connectors fully inserted?	No looseness	---
	Are the external wiring screws tight?	No looseness	---
	Are the connecting cables undamaged?	No damage	---

7-6-3 Replacing Faulty Units

Replace a faulty User Defined CAN Unit as soon as possible. Refer to the *DeviceNet Slaves Operation Manual* (W347) or the *MULTIPLE I/O TERMINALs Operation Manual* (W348) for details on replacing slaves. We recommend having spare Units available to restore network operation as quickly as possible.

Precautions

Observe the following precautions when replacing a faulty Unit.

- After replacement make sure that there are no errors with the new Unit.
- When a Unit is being returned for repair, attach a sheet of paper detailing the problem and return the Unit to your OMRON dealer.
- If there is a faulty contact, try wiping the contact with a clean, lint-free cloth dampened with alcohol.

Note To prevent electric shock when replacing a Unit, be sure to stop communications in the network and turn OFF the power supplies to all of the nodes (master and slaves) before removing the faulty Unit.

Appendix A
J1939 Protocol for Trucks and Busses

Society of Automotive Engineers

The Society of Automotive Engineers has more than 84,000 members - engineers, business executives, educators, and students from more than 97 countries - who share information and exchange ideas for advancing the engineering of mobility systems.

SAE J1939 Standards Collection

The Society of Automotive Engineers (SAE) Truck and Bus Control and Communications Subcommittee has developed a family of standards concerning the design and use of devices that transmit electronic signals and control information among vehicle components. SAE J1939 and its companion documents have quickly become the accepted industry standard and the Controller Area Network (CAN) of choice for off-highway machines in applications such as construction, material handling, and forestry machines.

The SAE J1939 Standards Collection is based on the Controller Area Network (CAN) data link layer (ISO 11898-1) using the extended frame format (29-bit identifiers). Several documents have undergone revision after the initial publication in 1998. The specifications have been added to, parts taken off and clarified. The set of specifications, available from SAE (www.sae.org), includes:

- J1939 Recommended Practice for a Serial Control and Communications Vehicle Network
- J1939/01 Recommended Practice for Control And Communications Network for On-Highway Equipment
- J1939/11 Physical Layer - 250k bits/s, Twisted Shielded Pair
- J1939/13 Off-Board Diagnostic Connector
- J1939/15 Reduced Physical Layer, 250K bits/sec., Un-Shielded Twisted Pair (UTP)
- J1939/21 Data Link Layer
- J1939/31 Network Layer
- J1939/71 Vehicle Application Layer
- J1939/73 Application Layer - Diagnostics
- J1939/75 Application Layer-Generator Sets and Industrial
- J1939/81 Network Management

The naming of the layers is not always compliant to the OSI reference model and to CiA's recommend terminology. The J1939/21 and J1939/31 define partly an application layer, and the J1939/71 and J1939/73 specify an application profile.

As a result of the standardization modern trucks have their diesel-engines, retarders, gearboxes and other equipment communicating with each other using this protocol.

Messages in J1939

Most messages on J1939 are intended to be broadcasts. This means that the data is transmitted on the network without directing them to a specific destination. This permits any device to use the data without requiring additional request messages. This also permits future software revisions to easily accommodate new devices (address assignments).

When a message must be directed to a particular device, a specific destination address can be included within the identifier of the message. An example of this is the transmission commanding a specific torque value from the engine vs. a specific torque value from a retarder. Requiring a destination must be considered when the message is first defined and published by the SAE committee because the destination address message format (PDU1) is slightly different from the broadcast message format (PDU2). PDU stands for Protocol Data Unit (i.e. Message Format).

CAN EXTENDED FRAME FORMAT	SOF	IDENTIFIER 11 BITS											SRR	IDE	IDENTIFIER EXTENSION 18 BITS																	RTR
J1939 FRAME FORMAT	SOF	PRIORITY			R	DP	PDU FORMAT (PF) 6 BITS (MSB)						SRR	IDE	PF (CONT.)		PDU SPECIFIC (PS) DEST. ADDRESS, GROUP EXT. OR PROPRIETARY								SOURCE ADDRESS								RTR
		3	2	1			8	7	6	5	4	3			2	1	8	7	6	5	4	3	2	1	8	7	6	5	4	3	2	1	
J1939 FRAME BIT POSITION	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33
CAN 29 BIT ID POSITION		28	27	26	25	24	23	22	21	20	19	18			17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	

J1939 uses the 29 bit identifier defined within the CAN protocol shown in figure above. The SOF, SRR, IDE and RTR bits will be ignored in the following description.

The first 3 bits are used for determining message priority during the arbitration process. A value of 000 has the highest priority. Higher priority messages would typically be used for high speed control messages. An example of this is the torque control message from the transmission to the engine. A lower priority would be used for data which is not time critical. An example of this vehicle road speed. The priority field should be programmable for each DC value so that network tuning can be performed by an OEM if necessary.

The next bit of the identifier is reserved. The bit should be set to 0 for transmitted messages. This default will permit future use of the bit for other purposes as defined by the SAE committee.

The next set of 9 bits in the identifier is the Data Content (DC) field. The DC field identifies what data is contained within the message. The first bit is used as a page selector. The later 8 bits (PDU Format) provide a page of 256 different values. Page 0 is intended to contain all the messages which are presently being defined. Page 1 is intended to provide additional expansion capacity for the future. Data Content values are described in J1939/71

The next 8 bits of the identifier are PDU specific (dependent on the message format). If the data content value is between 0 and 239,this field contains a destination address (PDU1). If the data content field is between 240 and 255, this field contains an extended data content (PDU2). This provides a larger set of values to identify different broadcast data.

The last 8 bits of the identifier contain the address of the device transmitting the message. For a given network, every address must be unique (256 available). This means that two different devices (ECUS) cannot use the same address.

Engine speed example

Now we want to control the engine speed of a diesel-engine. In specification J1939/71 we can find that we need for this the Parameter Group Number Torque/Speed Control #1:TSC1

The TORQUE/SPEED CONTROL #1: TSC1 has the following parameters:

Transmission repetition rate: when active; 10 ms to the engine - 50 ms to the retarder

Data length: 8 bytes

Data page: 0

PDU format: 0

PDU specific: Destination address

Default priority: 3

Parameter group number: 0 (00000016)

Byte:

1 Control bits Bit: 8-7 Not defined

6,5 Override control mode priority

4,3 Requested speed control conditions

2,1 Override control modes

2,3 Requested speed/Speed limit

4 Requested torque/Torque limit

5-8 Not defined

With this information we can compose the identifier to send with and where the requested speed should go in the data-field.

We use as source address the preferred address for a Management Computer, which is 39 (=27hex). But actually any address could do. The destination address is to engine#1 which has the preferred address 0.

So the identifier will look like this:

Priority + DP	PDU Format	PDU Specific	Source Address
0C	00	00	27

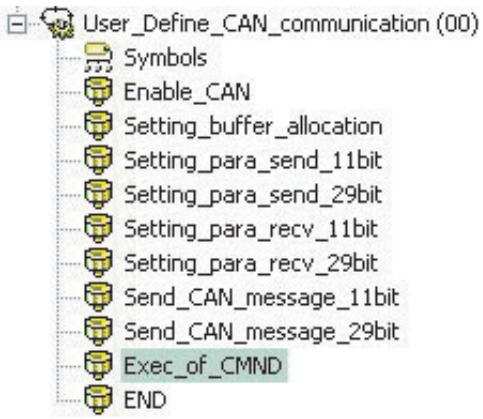
And the data will look like this:

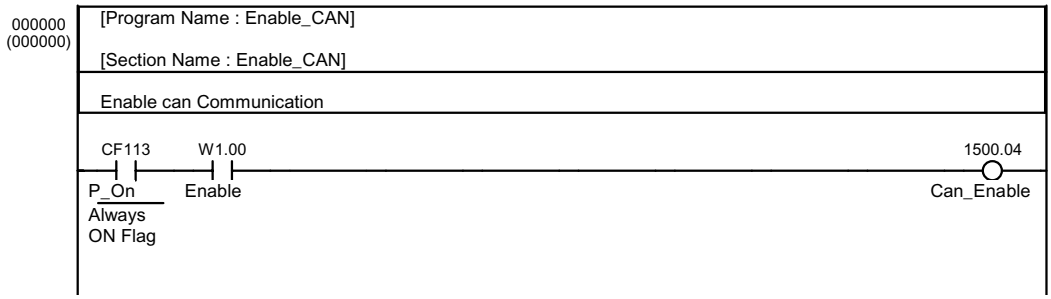
Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
00	Speed high byte	Speed low byte	00	FF	FF	FF	FF

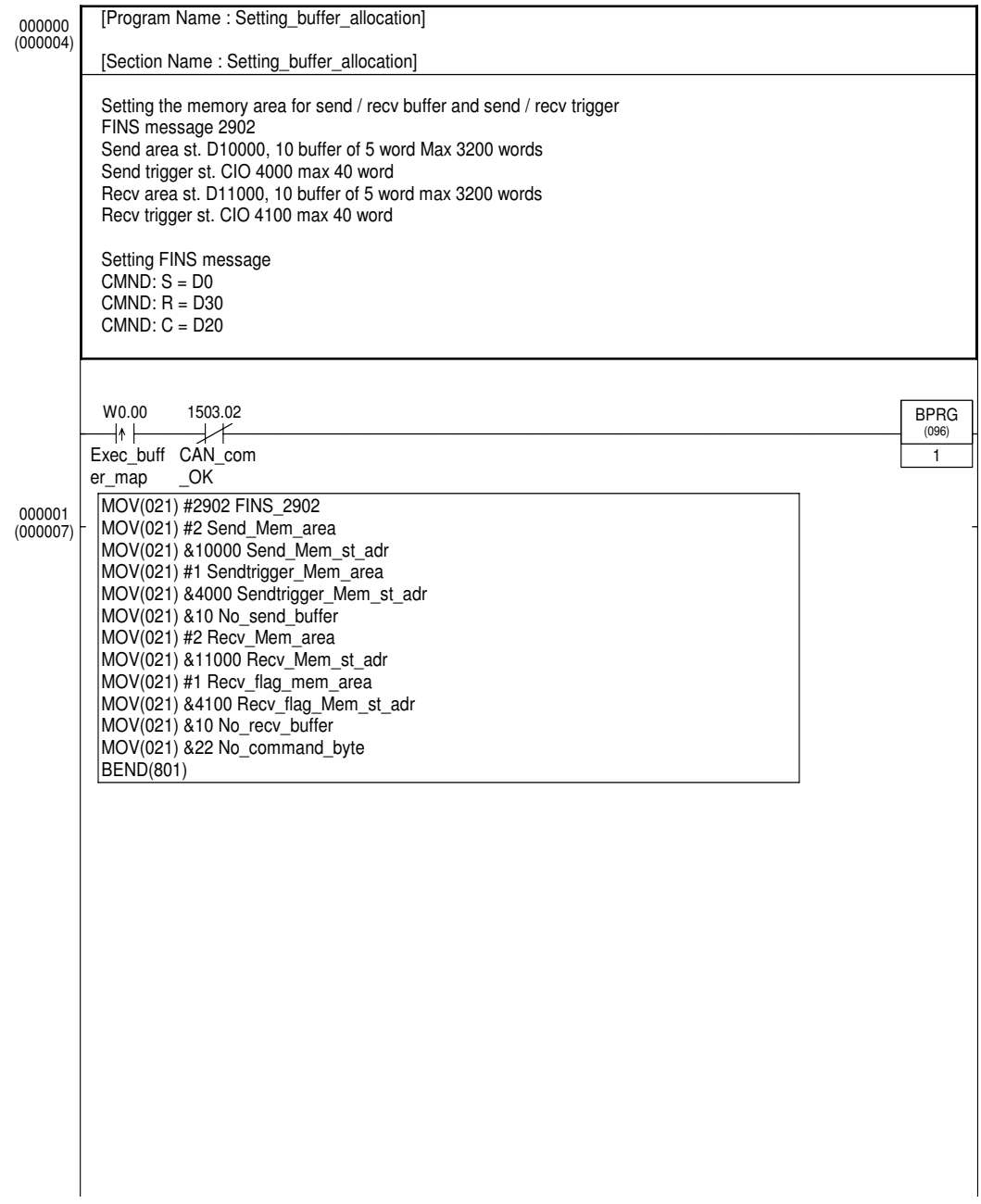
Typically in J1939 all 8 data-bytes of a CAN-message are send though they are not defined. There value should be then FF_{hex}.

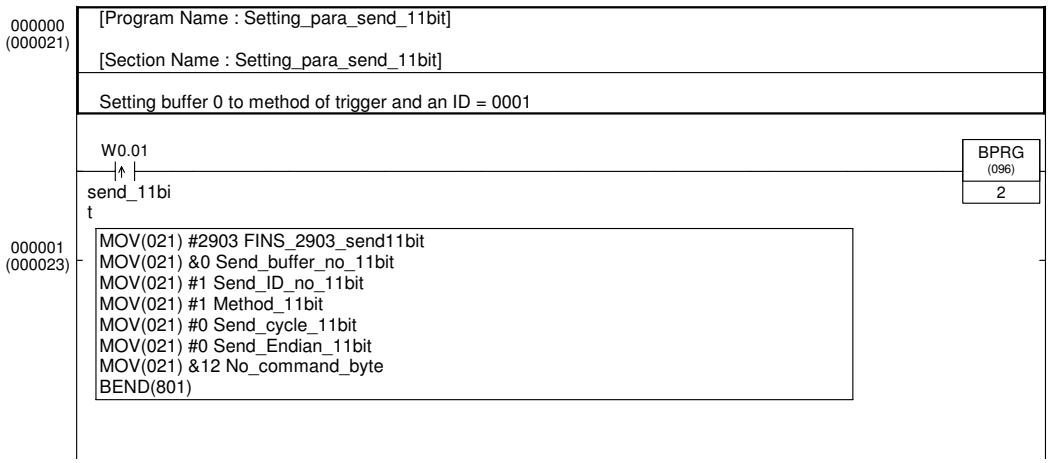
Appendix B
Programming examples

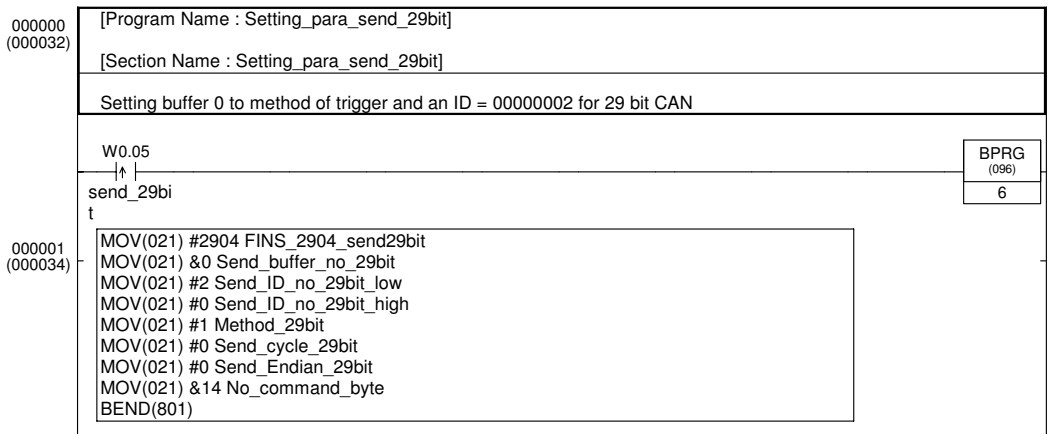
This example program shows how to configure the input and output buffers for the User Defined CAN Unit using FINS commands. The ladder logic shown in this appendix is ordered according to the PLC program lay-out as given in the figure below.

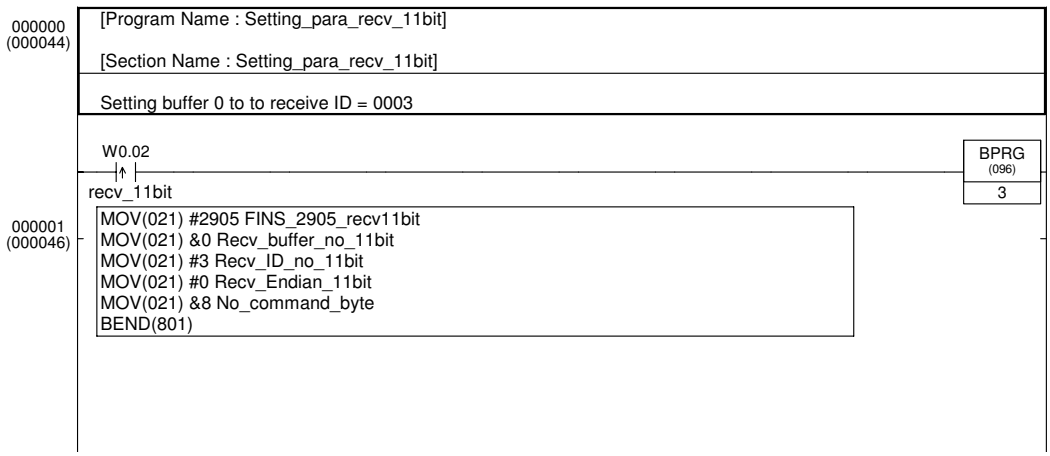


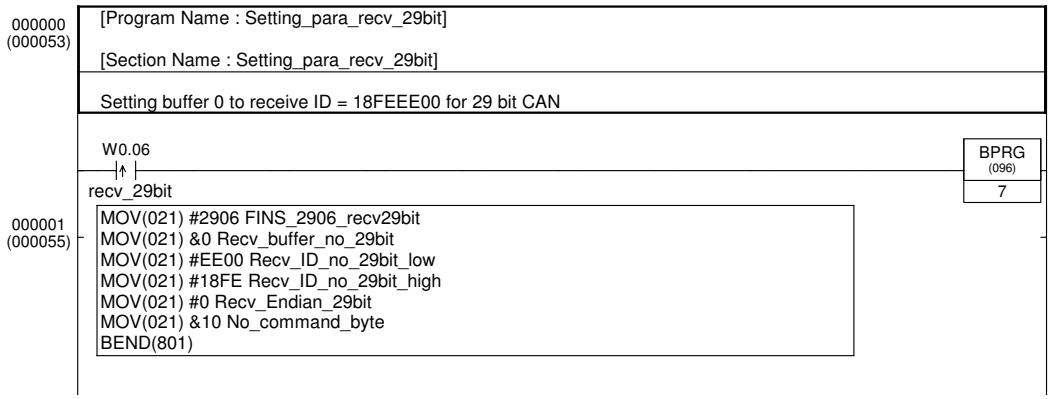

000000
(000000)
[Program Name : Enable_CAN]
[Section Name : Enable_CAN]
Enable can Communication
CF113
P_On
Always
ON Flag
W1.00
Enable
1500.04
Can_Enable

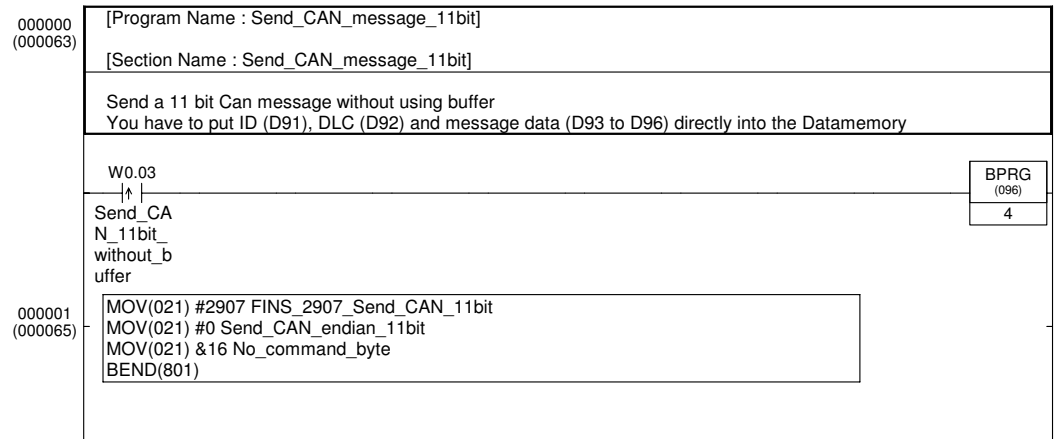

000000
(000004)
[Program Name : Setting_buffer_allocation]
[Section Name : Setting_buffer_allocation]
Setting the memory area for send / recv buffer and send / recv trigger
FINS message 2902
Send area st. D10000, 10 buffer of 5 word Max 3200 words
Send trigger st. CIO 4000 max 40 word
Recv area st. D11000, 10 buffer of 5 word max 3200 words
Recv trigger st. CIO 4100 max 40 word
Setting FINS message
CMND: S = D0
CMND: R = D30
CMND: C = D20
W0.00
Exec_buff
er_map
1503.02
CAN_com
_OK
BPRG
(096)
1
000001
(000007)
MOV(021) #2902 FINS_2902
MOV(021) #2 Send_Mem_area
MOV(021) &10000 Send_Mem_st_adr
MOV(021) #1 Sendtrigger_Mem_area
MOV(021) &4000 Sendtrigger_Mem_st_adr
MOV(021) &10 No_send_buffer
MOV(021) #2 Recv_Mem_area
MOV(021) &11000 Recv_Mem_st_adr
MOV(021) #1 Recv_flag_mem_area
MOV(021) &4100 Recv_flag_Mem_st_adr
MOV(021) &10 No_recv_buffer
MOV(021) &22 No_command_byte
BEND(801)

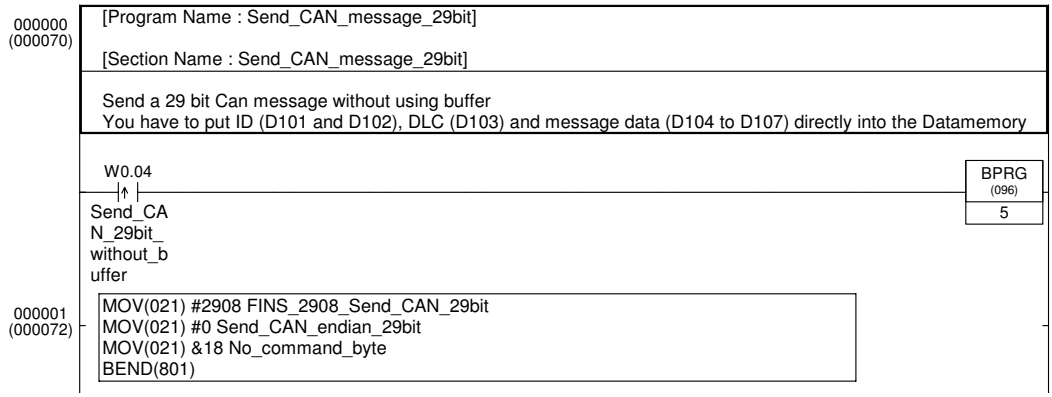

000000
(000021)
[Program Name : Setting_para_send_11bit]
[Section Name : Setting_para_send_11bit]
Setting buffer 0 to method of trigger and an ID = 0001
W0.01
send_11bi
t
BPRG
(096)
2
000001
(000023)
MOV(021) #2903 FINS_2903_send11bit
MOV(021) &0 Send_buffer_no_11bit
MOV(021) #1 Send_ID_no_11bit
MOV(021) #1 Method_11bit
MOV(021) #0 Send_cycle_11bit
MOV(021) #0 Send_Endian_11bit
MOV(021) &12 No_command_byte
BEND(801)

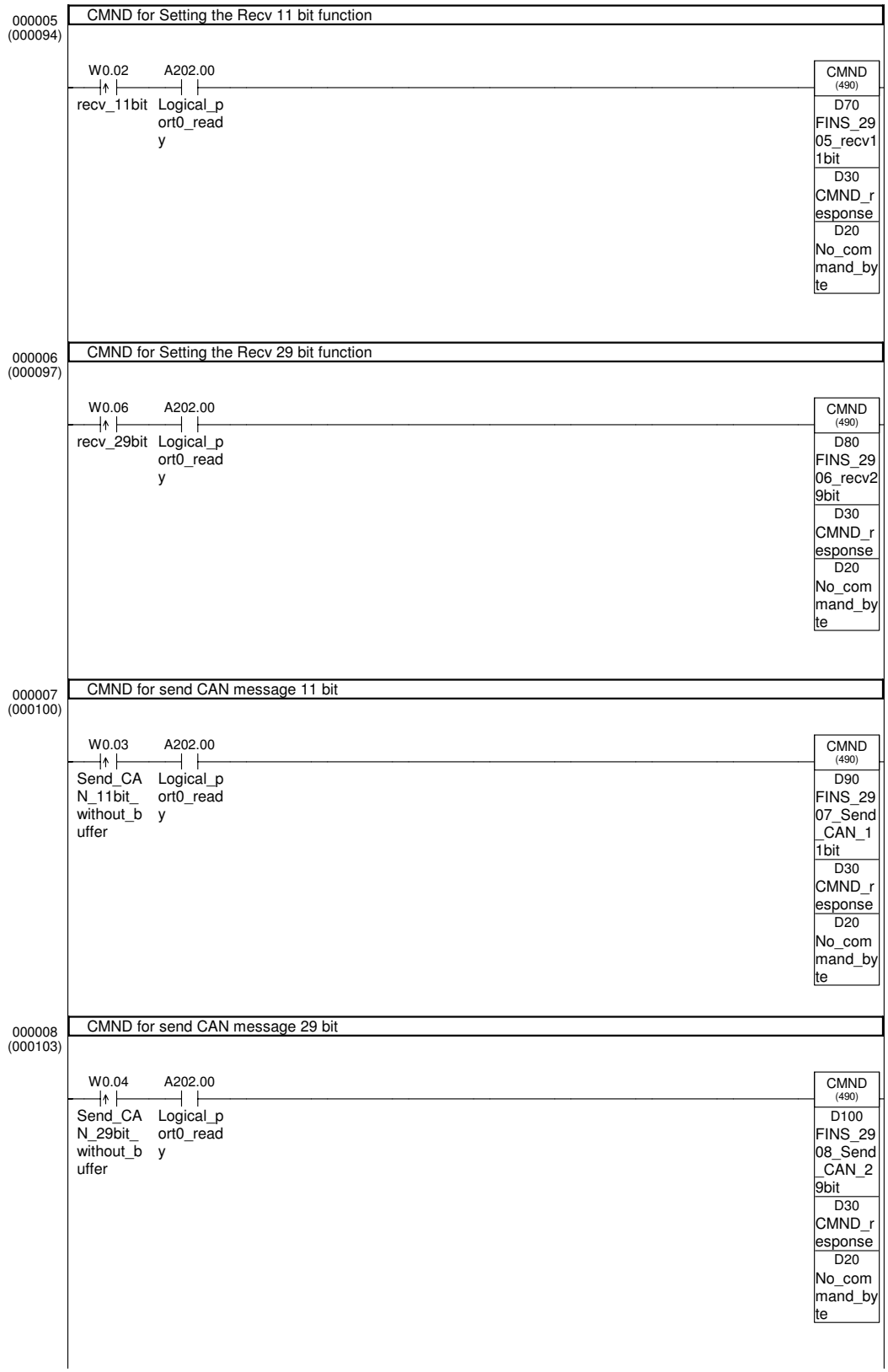

000000
(000032)
[Program Name : Setting_para_send_29bit]
[Section Name : Setting_para_send_29bit]
Setting buffer 0 to method of trigger and an ID = 00000002 for 29 bit CAN
W0.05
send_29bi
t
BPRG
(096)
6
000001
(000034)
MOV(021) #2904 FINS_2904_send29bit
MOV(021) &0 Send_buffer_no_29bit
MOV(021) #2 Send_ID_no_29bit_low
MOV(021) #0 Send_ID_no_29bit_high
MOV(021) #1 Method_29bit
MOV(021) #0 Send_cycle_29bit
MOV(021) #0 Send_Endian_29bit
MOV(021) &14 No_command_byte
BEND(801)

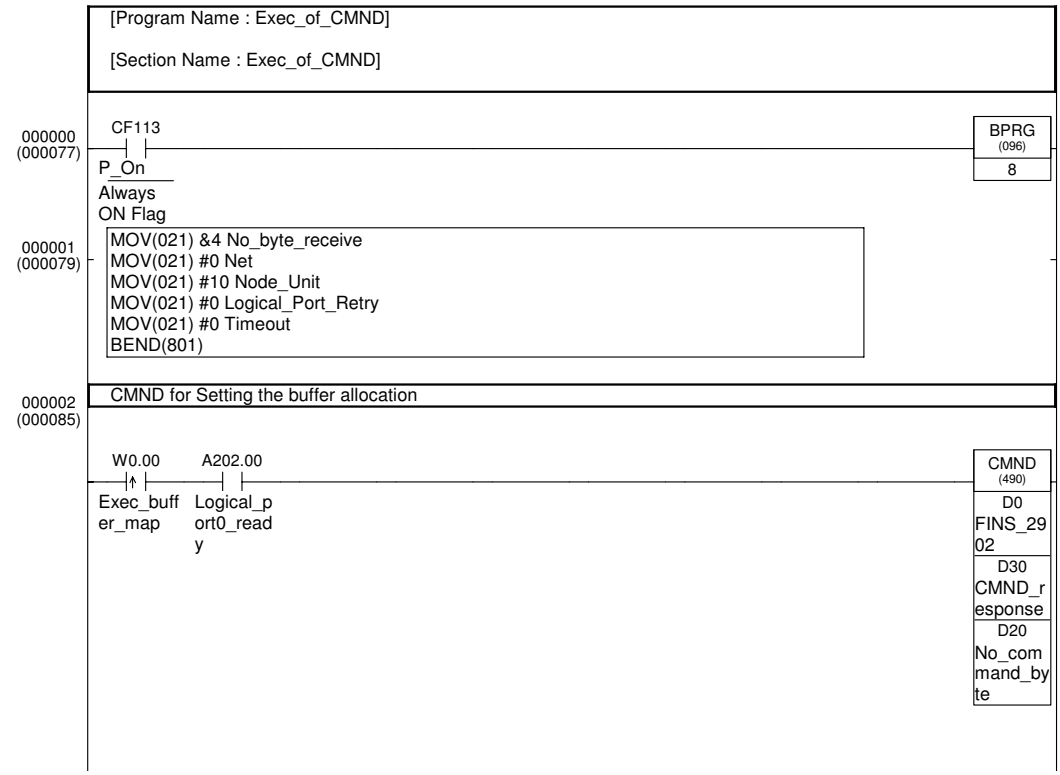

000000
(000044)
[Program Name : Setting_para_recv_11bit]
[Section Name : Setting_para_recv_11bit]
Setting buffer 0 to to receive ID = 0003
W0.02
recv_11bit
BPRG
(096)
3
000001
(000046)
MOV(021) #2905 FINS_2905_recv11bit
MOV(021) &0 Recv_buffer_no_11bit
MOV(021) #3 Recv_ID_no_11bit
MOV(021) #0 Recv_Endian_11bit
MOV(021) &8 No_command_byte
BEND(801)

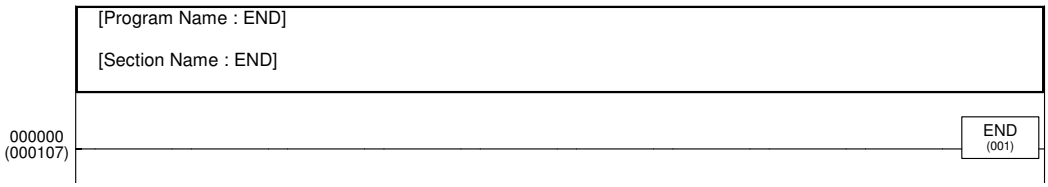

000000
(000053)
[Program Name : Setting_para_recv_29bit]
[Section Name : Setting_para_recv_29bit]
Setting buffer 0 to receive ID = 18FEEE00 for 29 bit CAN
W0.06
recv_29bit
BPRG
(096)
7
000001
(000055)
MOV(021) #2906 FINS_2906_recv29bit
MOV(021) &0 Recv_buffer_no_29bit
MOV(021) #EE00 Recv_ID_no_29bit_low
MOV(021) #18FE Recv_ID_no_29bit_high
MOV(021) #0 Recv_Endian_29bit
MOV(021) &10 No_command_byte
BEND(801)


000000
(000063)
[Program Name : Send_CAN_message_11bit]
[Section Name : Send_CAN_message_11bit]
Send a 11 bit Can message without using buffer
You have to put ID (D91), DLC (D92) and message data (D93 to D96) directly into the Datamemory
W0.03
Send_CA
N_11bit_
without_b
uffer
BPRG
(096)
4
000001
(000065)
MOV(021) #2907 FINS_2907_Send_CAN_11bit
MOV(021) #0 Send_CAN_endian_11bit
MOV(021) &16 No_command_byte
BEND(801)


000000
(000070)
[Program Name : Send_CAN_message_29bit]
[Section Name : Send_CAN_message_29bit]
Send a 29 bit Can message without using buffer
You have to put ID (D101 and D102), DLC (D103) and message data (D104 to D107) directly into the Datamemory
W0.04
Send_CA
N_29bit_
without_b
uffer
BPRG
(096)
5
000001
(000072)
MOV(021) #2908 FINS_2908_Send_CAN_29bit
MOV(021) #0 Send_CAN_endian_29bit
MOV(021) &18 No_command_byte
BEND(801)


000005
(000094)
CMND for Setting the Recv 11 bit function
W0.02
recv_11bit
A202.00
Logical_port0_ready
CMND (490)
D70
FINS_2905_recv11bit
D30
CMND_response
D20
No_command_byte
000006
(000097)
CMND for Setting the Recv 29 bit function
W0.06
recv_29bit
A202.00
Logical_port0_ready
CMND (490)
D80
FINS_2906_recv29bit
D30
CMND_response
D20
No_command_byte
000007
(000100)
CMND for send CAN message 11 bit
W0.03
Send_CAN_11bit_without_buffer
A202.00
Logical_port0_ready
CMND (490)
D90
FINS_2907_Send_CAN_11bit
D30
CMND_response
D20
No_command_byte
000008
(000103)
CMND for send CAN message 29 bit
W0.04
Send_CAN_29bit_without_buffer
A202.00
Logical_port0_ready
CMND (490)
D100
FINS_2908_Send_CAN_29bit
D30
CMND_response
D20
No_command_byte


[Program Name : Exec_of_CMND]
[Section Name : Exec_of_CMND]
000000
(000077)
CF113
P_On
Always
ON Flag
BPRG
(096)
8
000001
(000079)
MOV(021) &4 No_byte_receive
MOV(021) #0 Net
MOV(021) #10 Node_Unit
MOV(021) #0 Logical_Port_Retry
MOV(021) #0 Timeout
BEND(801)
000002
(000085)
CMND for Setting the buffer allocation
W0.00
Exec_buffer_map
A202.00
Logical_port0_ready
CMND
(490)
D0
FINS_2902
D30
CMND_response
D20
No_command_byte


[Program Name : END]
[Section Name : END]
000000
(000107)
END
(001)

Abbreviations

Abbreviation	Description
A - C	
CAN	Controller Area Network.
CIO	Control Input and Output area in CPU of PLC system
CPU	Central Processor Unit, in PLC system.
CRC	Cyclic Redundancy Check
D	
DM	Data Memory in CPU of PLC system
E	
EC	Electromagnetic Compatibility
EM	Extended Memory
EOF	End Of Frame
EVx	Event x for the User Defined CAN Unit EV1 = Memory areas configured EV2 = at least one CAN message configured EV3 = Enable communications EV4 = Disable communications EV5 = Bus off
F - G	
FA	Factory Automation.
FINS	Factory Intelligent Network System.
H	
HLP	Higher Layer Protocol
I - K	
I/O	Input and Output.
IC	Integrated Circuit
IDE	CAN Protocol bit definition: Identifier Extension Bit
ISO	International Organization for Standardization.
L	
LLC	Logical Link Control, OSI layer.
LSI	Large Scale Integration
M	

Abbreviation	Description
MAC	Medium Access Control, OSI layer.
N	
NRZ	Non Return to Zero
O	
OSI	Open System Interconnection.
P - Q	
PC	Personal Computer
PCB	Printed Circuit Board
PDU	Protocol Data Unit
PLC	Programmable Logic Controller.
R	
RTR	CAN Protocol bit definition: Remote Transmission Request
S - Z	
SEA	Society of Automotive Engineers
SMx	Send Mode x, used send mode for CAN messages. SM1 = Triggered SM2 = On change SM3 = Cyclic
SOF	CAN Protocol bit definition: Start Of Frame
SOF	Start Of Frame
SRR	CAN Protocol bit definition: Substitute Remote Request
STx	State x for the User Defined CAN Unit. ST1 = Initialising ST2 = Start-up error ST3 = Not configured ST4 = Configured ST5 = Communicating
STP	Shielded Twisted Pair
UTP	Unshielded Twisted Pair

Index

Numbers

29-bit identifier, 32

A

About
  Manual, ix
Ack Field, 29
Acknowledge, 16
Addressing scheme, 25
Allocated CIO and DM Words
  Manual, 35
Application areas, 2
Application Precautions
  Manual, xiv
Arbitration field, 29

B

Basic Operating Procedure, 9
Bit level errors, 30
Bitwise arbitration, 27
Bosch
  Robert Bosch FmbH, vi
Bus access, 27
Bus access control, 28
Bus allocation, 27, 28

C

Cable length, 4
CAN, 2
  Cable length, 4
  Configuration, 7
  Connector, 20
  Controller Area Network, 24
  Data frame, 5
  Identifier, 6
  LLC, 5
  Messages, 5
  network, 6, 54
  Reliability, 31
  Residual error, 32
  Serial bus, 3
  Transmission speed, 3
  Troubleshooting, 7
CAN Operation
  Manual, 23
CIO, 14, 16
CIO Area, 8, 37
CJ1W CORT21, 6
CJ1W-CORT21, ix, 12
CMND, 51, 52
Command Codes, 49
Communication enable, 39
Communications flag, 52
Configuration, 7, 21
Conformance to EC Directives
  Manual, xvi
Connector, 20
Control Field, 29
Controller Area Network, 2, 24
CPU, xiii, xv, 7, 19
CRC Field, 29
Cycle time, 70
Cyclic refresh, 14

D

Data
  exchange, 25
Data field, 29
Data frame, 5
Declaration of Confirmity, xvi
Destructive bus allocation, 27
Dimensions, 21
DIP switch, 19
DM Area, 7, 8, 37
Dot indicators, 17, 18
dot indicators
  See also seven-segment display

E

EMC Directives, xvi
ERR, 17
Error
  ACK errors, 30
  Cyclic Redundancy Check, 30
  Detection, 30
  Frame check, 30
Error flag, 30
Error frame, 16
EV1, 14
EV2, 14
EV3, 14
EV4, 14
EV5, 14
Events, 14
Extended format, 32

F

Factory Automation, xii
Features and Specifications
  Manual, 1
FINS, 7, 8, 48, 50
  2902, 54
  2903, 57
  2904, 58
  2905, 60
  2906, 62
  2907, 63
  2908, 64
  2909, 66
  Command Codes, 49
  Response Codes, 49
FINS Commands and Responses
  Manual, 47

G-H

General Precautions
  Manual, xii

I

I/O allocations, 8
I/O functions, 33
I/O refresh time, 70
I/O response time, 74
IDE bit, 33
Identifier, 6, 29
Indicators, 12, 17
Input buffer, 14, 44
Installation, 9
Intermediate buffer, 33
Intermission, 30
ISO 11898, 25

L-M

LED, 12
License of CAN, 2
LLC, 5
LLC sublayer, 25
Low voltage directive, xvi
MAC sublayer, 24
Manual
  About, ix
  Allocated CIO and DM Words, 35
  Application Precautions, xiv
  CAN Operation, 23
  Conformance to EC Directives, xvi
  Features and Specifications, 1
  FINS Commands and Responses, 47
  General Precautions, xii
  Nomenclature and Installation, 11
  Operating Environment Precautions, xiii
  Operation, 69
  Precautions, xi
  Safety Precautions, xii
  Troubleshooting and Maintenance, 79
Message delay time, 73
Message frame formats, 29
Messages, 5
Mounting, 21

N

network, 6, 54
Nomenclature and Installation
  Manual, 11
Non-destructive bus access, 27
Number of delayed send messages, 41

O-P

Object storage, 33
OMRON
  Copyright notice, vi
Operating Environment Precautions
  Manual, xiii
Operation
  Manual, 69
OSI Reference Model, 24
OSI reference model
  ISO 11898, 3
  ISO-7498, 2
  Layer 1, Physical Layer, 2
  Layer 2, Data Link Layer, 2
Output buffer, 14, 43
Performance, 70, 74
Physical Layer, 24
PLC cycle time, 70
Precautions
  Manual, xi
Priority, 26
Program execution time, 75
programming
  See also instructions
  See also ladder programs

R

Receive Flags, 14
Receive Flags Area, 43

Receiving CAN messages, 16
Reception of CAN messages, 77
Reliability, 31
Residual error, 32
Response Codes, 49
Re-transmission, 30
Rotary switches, 19
RTR bit, 33
RUN, 17

S

SAE
  The Society of Automotive Engineers, vi
Safety Precautions
  Manual, xii
SEND, 52
Send mode, 15
Send Trigger Area, 42
Send Triggers, 14
Sending CAN messages, 15
Sending messages, 25
Serial bus, 2, 3
Seven-segment Digits, 18
Seven-Segment Display, 17
SM1, 15, 43
SM2, 15
SM3, 15
ST1, 13
ST2, 13
ST3, 13
ST4, 13
ST5, 13
Start-up, 18
States, 12
Status communication, 40
Status Indicators, 17
Switch, 19
Switch settings, 19

T

timing
  communications, 69
Transmission of CAN messages, 76
Transmission speed, 3
Troubleshooting, 7
troubleshooting, 79
Troubleshooting and Maintenance
  Manual, 79

U

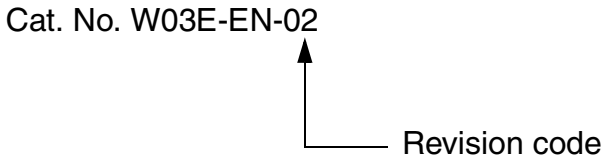
Unit
  General specifications, 7
  states, 12
Unit addressing, 25
Unit Cyclic refresh, 14
User Defined CAN Unit, 6

V-W

Word allocations, 36

Revision History

A manual revision code appears as a suffix to the catalog number on the front cover of the manual.



The following table outlines the changes made to the manual during each revision.

Revision Code	Date	Revised Content
01	August 6, 2004	Initial version.
02	November 2, 2004	Bugs fixed